

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Luka Krsnik

Družabno omrežje podjetij

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: dr. Andrej Brodnik

Ljubljana 2014

Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil L^AT_EX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Socialna omrežja na spletu so resničnost in pogosta oblika sodelovanja med posamezniki. Na nek način tvorijo eko-sistem posameznikov. Na podoben način bi lahko vzpostavili prostovoljni eko-sistem podjetij, preko katerih bi si le-ta izmenjavala zahteve in ponudbe za posle.

V diplomski nalogi oblikujte predlog eko-sistema podjetij, preko katerega le-ta lahko sodelujejo in si izmenjujejo ponudbe in povpraševanja za različne izdelke in storitve. Poleg same zasnove sistema predlagajte primerne tehnologije za njegovo izvedbo in prototipno implementirajte del sistema.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Luka Krsnik, z vpisno številko **63110179**, sem avtor diplomskega dela z naslovom:

Družabno omrežje podjetij

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom dr. Andreja Brodnika,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 15. septembra 2014

Podpis avtorja:

Zahvaljujem se mentorju, dr. Andreju Brodniku za svetovanje ter usmerjanje pri izdelavi diplomskega dela in za ponujeno priložnost za sodelovanje z Žigom Veselom. Ob tem bi se zahvalil še Žigu za pomoč in podjetniško idejo, brez katere te diplome nebi bilo. Zahvalil bi se tudi družini in prijateljem, ki so mi stali ob strani vsa leta študija, mi pomagali, ko sem potreboval pomoč, in me spodbujali, ko sem potreboval spodbudo.

Diplomsko delo posvečam Mateji in Gregu,
bodočima nevesti in ženinu.

Kazalo

Povzetek

Abstract

1	Uvod	1
1.1	Motivacija	1
1.2	Definicija problema	3
1.3	Struktura naloge	11
2	Obstoječe rešitve	13
2.1	Bolha	13
2.2	eBay	14
2.3	Europages	15
2.4	Ostalo	16
3	Uporabljene tehnologije	17
3.1	PostgreSQL	17
3.2	Python	18
3.3	JavaScript	20
3.4	Spletno programiranje	22
4	Opis rešitve	25
4.1	Arhitektura aplikacije	25
4.2	Procesni diagram	28
4.3	Entitetno relacijski model	39

KAZALO

5	Ovrednotenje rešitve	47
5.1	Funkcionalnost	47
5.2	Varnost	62
6	Zaključek	63
A	Podatkovni model podjetniške ideje	65

Seznam uporabljenih kratic

kratica	angleško	slovensko
B2B	<i>business to business</i>	poslovanje med podjetji
B2C	<i>business to consumer</i>	poslovanje podjetij s porabniki
C2C	<i>consumer to consumer</i>	poslovanje med porabniki
SQL	<i>Structured Query Language</i>	strukturiran povpraševalni jezik
XML	<i>EXtensible Markup Language</i>	razširljiv označevalni jezik
SOAP	<i>Simple Object Access protocol</i>	preprosti objektno dostopni protokol
MVC	<i>model view controller</i>	model pogled nadzor
JSON	<i>JavaScript Object Notation</i>	JavaScriptova objektna notacija
REST	<i>Representational State Transfer</i>	reprezentativen prenos stanja
HTML	<i>HyperText Markup Language</i>	jezik za označevanje nadbesedila
AJAX	<i>Asynchronous JavaScript and XML</i>	asinhorni JavaScript in XML
CSS	<i>Cascading Style Sheets</i>	kaskadne stilske podloge
JWT	<i>JSON Web Token</i>	spletni žeton tipa JSON
URL	<i>Uniform Resource Locator</i>	enolični krajevnik vira
DOM	<i>Document Object Model</i>	dokumentni objekt modela
CRUD	<i>Create Read Update Delete</i>	naredi beri posodobi izbriši
POJO	<i>Plain Old JavaScript Object</i>	čisti stari JavaScript objekt
TLS	<i>Transport Layer Security</i>	kriptiran protokol za omrežno varnost

Povzetek

V diplomski nalogi je opisan začetek implementacije podjetniške ideje. Gre za spletno stran, s pomočjo katere je možno razširiti, poenostaviti in pohitriti klasično poslovanje ter s tem olajšati delo podjetij v fazi iskanja in sklepanja poslov. Za to smo se odločili, ker trenutno obstajajo le aplikacije, ki se delno prekrivajo z našo rešitvijo, nobena izmed njih pa ni tako celovita in osredotočena na isto ciljno publiko.

Za izdelavo smo uporabili 3-slojno arhitekturo. Najpomembnejši del predstavitvenega sloja predstavlja JavaScriptovo ogrodje AngularJS, na logičnem delu je to Pythonovo ogrodje Django REST Framework, na podatkovnem sloju pa PostgreSQL.

V okviru diplomske naloge je izdelano okostje spletne aplikacije, ki pokriva glavne funkcionalnosti. Tako je narejena administracija uporabnikov, pregled in upravljanje s podjetji, možnost objave povpraševanj in ponudb ter možnost pisanja in sklepanja pogodb.

Ključne besede: podjetništvo, spletna stran, 3-slojna arhitektura, JavaScript, AngularJS, Python, Django REST Framework, PostgreSQL.

Abstract

The thesis contains a description about starting the implementation of business idea. We made a web page, which helps in extending, simplifying and accelerating classical business, making it easier for companies in phase of searching and concluding business. We decided to make such application, because at the moment, up to our knowledge, application, which would comprehensively cover this functionality doesn't exist. There are only some applications that focus on other segments of population or they cover the required functionalities partially.

During developement of an application we used 3-tier architecture. For the most important part of presentation tier is used JavaScripts framework AngularJS, whereas on logical tier the role was taken by Python's framework Django REST Framework. Data tier's most notable member is PostgreSQL.

In the thesis we narrowed the functionality down to skeleton of the final application. It contains administration of users, overview and management of companies and ability to post inquiries and offers. It also covers writing and concluding contracts.

Keywords: business, web page, 3-tier architecture, JavaScript, AngularJS, Python, Django REST Framework, PostgreSQL.

Poglavje 1

Uvod

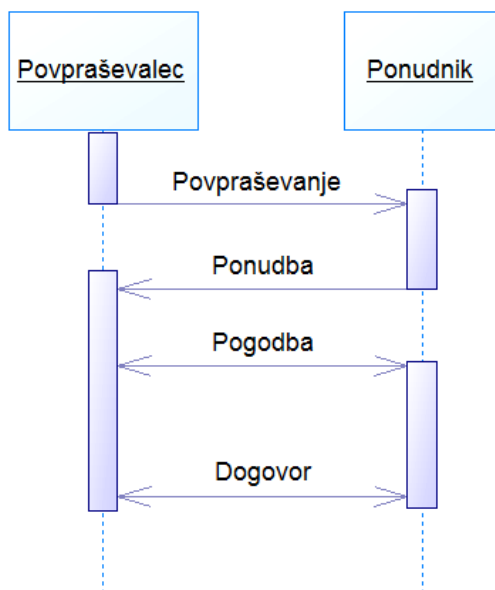
1.1 Motivacija

Tvorec poslovne ideje je Žiga Vesel. Sodeloval je tudi pri prepoznavanju ključnih funkcionalnosti in projektu prispeval predvsem domensko specifično znanje. Naša naloga je bila tehnična izvedba.

Diplomska naloga je začetni del tehničnega razvoja poslovne ideje. Vsebuje spletni sistem za lajšanje dela podjetij pri iskanju in sklepanju poslov. Glavni cilj spletne strani je izgradnja skupnosti, v kateri bi bilo medsebojno povezovanje podjetij enostavno. To je pomembno zato, da imajo ta pri povpraševanjih več kot samo eno ponudbo in posledično večjo pogajalsko moč. Pomembno je tudi v fazi iskanja dela, saj več povpraševanj pomeni lažje iskanje dela.

Klasično poslovanje (Slika 1.1) poteka tako, da neko podjetje - povpraševalec - navede storitev ali produkt, ki ga potrebuje, ter želje sporoči večim ponudnikom storitev ali produktov. Ti povpraševalskemu podjetju odgovorijo s ponudbo - morebitnim opisom izdelkov ali storitev. Podjetje, ki je prejelo več ponudb za delo, se nato odloči za najboljšega ponudnika, s katerim nadaljujeta sklepanje posla. Sledi usklajevanje podrobnosti ter zapis in podpis pogodbe. S tem se tudi zaključi faza dogovarjanja na katero smo osredotočeni v podjetniški ideji.

Aplikacijo smo razvili, ker smo spoznali, da se klasično poslovanje da razširiti, poenostaviti in pohitriti.



Slika 1.1: Shema klasičnega poslovanja.

Razširiti ga je mogoče z objavo na spletu, saj to pomeni, da so povpraševanja vidna večim potencialnim partnerjem, kar prinese več ponudb v krajšem časovnem obdobju.

Poenostavitev je možna zaradi konsistence prikaza, saj so vsa povpraševanja in ponudbe zasnovane na enak način, kar omogoča boljši pregled in lažje odločanje.

Večjo hitrost pa je možno doseči predvsem v fazi sklepanja pogodb. V tej fazi namreč povpraševalec in ponudnik spreminjata isti dokument in si ga med seboj izmenjujeta. Z objavo dokumenta na spletu je možno interaktivno spreminjanje z obeh strani hkrati.

Dolgoročni cilj aplikacije je izgradnja skupnosti, v katero bi bili vključeni podjetniki in drugi zastopniki podjetij. V načrtu je tudi izgradnja spletnega foruma, preko katerega si bodo ti izmenjavali mnenja, nasvete...

1.2 Definicija problema

Celotno funkcionalnost aplikacije lahko razdelimo na štiri obsežnejše dele in nekaj manjših. To so funkcionalnosti, ki se navezujejo na uporabnike, podjetja, povpraševanja, investicije in ostalo. Izgradnja aplikacije je narejena v enakem zaporedju, kot so navedene funkcionalnosti. Podjetja namreč ne morejo obstajati, če ne obstajajo uporabniki, povpraševanja in investicije pa so odvisne od podjetij, ki povprašujejo ali objavljajo investicijo.

Funkcionalnost uporabnikov zajema registracijo uporabnikov, prijavo in odjavo, pregled uporabnikov, iskanje uporabnikov, spremembe profila in administracijo uporabnikov. Na tem mestu so zelo pomembne tudi vloge uporabnikov. Obstajajo tri: vloga administratorja, urejevalca in navadnega uporabnika. Administrator skrbi za administracijo uporabnikov. Samo on lahko dostopa do seznama vseh uporabnikov in ureja tudi druge uporabnike (ne pa samo sebe). Urejevalci se od navadnih uporabnikov razlikujejo v tem, da lahko urejajo določene podstrani, kot sta na primer novice in pomoč. Pomembna razlika je še v tem, da lahko preverjajo pristojnosti navadnih uporabnikov v podjetjih.

V okviru funkcionalnosti podjetja so zajete registracija podjetja, pregled podjetij, urejanje podjetja in prijava ter potrjevanje k podjetju. Ker je pod enim podjetjem lahko aktivnih več uporabnikov, vsi ne smejo imeti vseh pooblastil. Tako ima uporabnik, ki ustvari podjetje, na aplikaciji vsa dovoljenja za upravljanje z njim, lahko pa jih dodeli tudi drugim. Ta pooblastila so možnost objave povpraševanja preko izbranega podjetja, možnost sprejetja povpraševanja, možnost objave ponudbe, možnost spreminjanja lastnosti podjetja in možnost spreminjanja pooblastil uporabnikov v okviru podjetja.

Funkcionalnost povpraševanj predstavlja vso funkcionalnost, ki se navezuje na povpraševanja, ponudbe in sklepanje pogodb. Sem spadajo objava povpraševanj, pregled povpraševanj, iskanje povpraševanj, objava ponudbe, pregled ponudb in sklepanje pogodb.

Funkcionalnost, ki je povezana z investicijami, zajema ponujanje investicij, seznam investicij, iskanje investicij, prijave na investicije in pregled prijav.

S tem pa seveda ni zajeta čisto vsa funkcionalnost, ki jo ponuja naša aplikacija.

Ta omogoča tudi urejanje in pregled različnih podstrani, kot so novice, kontakt, o podjetju in pomoč. Možno pa se je tudi pogovarjati z drugimi uporabniki na strani.

1.2.1 Uporabniške zgodbe

Za boljšo koordinacijo je celoten projekt razdeljen na več delov, konkretnije na uporabniške zgodbe. Te si sledijo v enakem zaporedju kot je predstavljena funkcionalnost v začetku tega podpoglavja 1.2. Pri pripravi tega dela diplomske naloge je sodeloval tudi idejni vodja Žiga Vesel.

Registracija uporabnika

Vsak neregistriran uporabnik ima možnost registracije. V tej fazi mora vpisati uporabniško ime, e-naslov, lastno ime, priimek in geslo.

Prijava

Vsak registriran uporabnik mora imeti možnost prijave, kjer sta potrebna uporabniško ime in geslo.

Odjava

Uporabnik ima možnost, da se odjavi in prijavi pod drugim imenom.

Pregled uporabnikov

Administrator lahko vidi listo vseh uporabnikov. Vidna so uporabniška imena, e-naslovi, lastna imena in priimki.

Filtriranje uporabnikov

Sistemi administrator, ki vidi seznam vseh uporabnikov, lahko filtrira uporabnike po uporabniškem imenu, e-naslovu, imenu ali priimku.

Spreminjanje osebnih podatkov

Vsak uporabnik ima možnost spreminjanja osebnih podatkov (uporabniško ime, e-naslov, ime, priimek in geslo). Pri spreminjanju gesla je potrebno vpisati tudi staro geslo.

Administracija uporabnikov

Administrator lahko drugim uporabnikom spremeni profil in pooblastila. Poleg navadnih uporabnikov obstajajo še urejevalci in administratorji.

Registracija podjetja

Podjetje lahko registrira katerikoli registriran uporabnik. Pri tem mora podati uradno ime podjetja, ki ga zastopa in njihov naslov. Možni so tudi vnosi panoge delovanja in morebiten dodaten opis ali informacije o podjetju. Za tem mora administrator imeti možnost, da potrdi oziroma zavrže določeno registracijo na podlagi pristojnosti uporabnika naše strani, znotraj dejanskega podjetja.

Seznam podjetij

Vsi navadni uporabniki lahko vidijo listo podjetij, ki so prijavljena na to spletno stran in potrjena. Poleg imena podjetja sta napisana tudi naslov in panoga dela. Administratorjem in urejevalcem so na tem mestu vidna še nepotrjena podjetja in informacija o tem, ali je bilo določeno podjetje že potrjeno.

Podrobnosti podjetja

Ko uporabnik to zahteva, so vidne podrobnosti podjetja (ime, naslov, dodatni opis, panoge delovanja). Vidno mora biti tudi koliko poslov je podjetje že izvedlo in kakšne so ocene podjetja. Tu lahko uporabnik začne tudi neformalni pogovor z odgovorno osebo podjetja. V tem primeru ga stran usmeri na pogovor, v katerem lahko enostavno dostopa do podjetja, ki povprašuje, in lastnega podjetja. Ko uporabnik dostopa do svojega (izbranega) podjetja, ima na voljo tudi hitre povezave do svojih povpraševanj in ponudb.

Iskanje podjetja

Spletna stran omogoča iskanje podjetja preko filtra. Išče se lahko po panogah, imenu in naslovu.

Prijava in potrjevanje k podjetju

Prijavljen uporabnik ima možnost, da se prijavi k podjetju, ki že obstaja. Po tem, ko to stori, ima pooblaščen član interesnega podjetja možnost, da prijavo sprejme oziroma zavrne. Če jo sprejme, mora uporabniku določiti še pristojnosti znotraj podjetja (te pristojnosti so: sprejemanje ponudb, prijavljanje na povpraševanja, objava povpraševanj, dodeljevanje pooblastil znotraj podjetja drugim, spreminjanje opisa profila podjetja, objava investicij, prijava na investicije in potrjevanje investicije).

Spreminjanje lastnosti podjetja

Vsak uporabnik, ki ima znotraj določenega podjetja za to pooblastila, ima možnost spreminjanja lastnosti podjetja (ime, naslov, tip podjetja, panoge delovanja, dodaten opis). Uporabnik z zadostnimi pooblastili mora imeti tudi možnost potrjevanja oziroma odstranjevanja drugih uporabnikov k podjetju in regulacijo njihovih pooblastil.

Objava povpraševanja

Pri objavi povpraševanja mora uporabnik, ki ima v okviru izbranega podjetja zadostna pooblastila, posredovati ime povpraševanja in datum, do katerega bo povpraševanje aktivno. Lahko navede še panogo povpraševanja, okraj povpraševanja in druge podrobnosti.

Seznam povpraševanj

Vsi prijavljeni uporabniki lahko vidijo listo povpraševanj, ki so objavljena na tej spletni strani. Vidno mora biti ime povpraševanja, ime povpraševalskega podjetja, lokacija povpraševanja, panoga po kateri se povprašuje in časovne omejitve.

Iskanje povpraševanja

Omogočeno je iskanje s filtriranjem po panogah, imenih povpraševanj, imenih podjetij in področjih povpraševanja.

Podrobnosti povpraševanja

Če uporabnik izbere neko povpraševanje, so vidne določene podrobnosti podjetja (ime, ocena in število izvedenih poslov) in informacije o izbranem povpraševanju (ime povpraševanja, časovne omejitve, panoge povpraševanja, morebitne dodatne informacije in področje povpraševanja). Tu lahko uporabnik začne tudi neformalni pogovor z odgovorno osebo podjetja, ki povprašuje. V tem primeru ga stran usmeri na pogovor, v katerem lahko enostavno dostopa do podjetja, ki povprašuje, lastnega podjetja in podrobnosti povpraševanja.

Prijava na povpraševanje

Za to pooblaščen uporabnik se lahko odloči, da se njegovo izbrano podjetje formalno prijavi na povpraševanje in sicer tako, da napiše podrobnosti ponudbe. Potem lahko povpraševalec vidi njegovo ponudbo.

Pregled ponudb

Povpraševalec lahko vidi vse ponudbe, ki se nanašajo na povpraševanje. Pregled je narejen preko imen podjetij, ki so objavila ponudbo.

Podrobnosti ponudbe

Povpraševalec s klikom na ponudniško podjetje lahko odpre podrobnosti neke ponudbe. Pri tem se pojavi besedilo ponudbe, možnost njenega sprejetja, vse lastnosti podjetij, ki objavljata ponudbo ali povpraševanje in lastnosti povpraševanja. Tudi tu lahko povpraševalec začne pogovor s ponudnikom (in obratno).

Spreminjanje ponudbe

Ponudnik (z zadostnimi pooblastili) lahko spreminja ponudbo vse dokler je povpraševalec ne sprejme.

Objava pogodbe

Potem, ko povpraševalec sprejme ponudbo, se vsebina le te prekopira v pogodbo. To lahko spreminjata tako povpraševalec kot ponudnik, dokler je obadva ne potr-dita. Ko se to zgodi, ni več mogoče spreminjati dokumenta. Aktivira se možnost za evalvacijo podjetja na drugi strani, ob tem pa se poveča še število opravljenih poslov.

Ponujanje investicij

Za to pooblaščen član podjetja lahko ponudijo investicijo. Navesti mora ime investicije in datum do katerega bo investicija na voljo. Opcijsko lahko doda podrobnosti investicije, okraj in panogo investiranja.

Seznam ponujenih investicij

Vsi prijavljeni uporabniki lahko vidijo listo investicij, ki so objavljene na tej spletni strani. Vidno mora biti pod katero panogo spada investicija, ime investitorja, področje investicije in časovne omejitve.

Iskanje investicij

Omogočeno je iskanje investicij preko filtriranja po panogah, okrajih, imenih investicije in podjetjih.

Podrobnosti investicij

Če uporabnik izbere neko investicijo, se mu morajo prikazati podrobnosti investicije (ime investicije, časovne omejitve, panoge investiranja in podrobnosti investicije) ter določene podrobnosti podjetja (ime, ocena in število izvedenih investicij). Če je uporabnik zainteresiran za investicijo, lahko začne pogovor z investitorjem. Lahko pa se tudi prijavi za investicijo.

Pregled prijav

Investitor lahko vidi vse prijave, ki se nanašajo nanj. Pregled je narejen preko imen podjetij, ki so se prijavila.

Podrobnosti prijav

Investitor s klikom na prijavljeno podjetje lahko odpre podrobnosti neke prijave. Pri tem se pojavi besedilo prijave, možnost njenega sprejetja in vse lastnosti podjetja, ki se prijavlja. Investitor lahko začne tudi pogovor s prijavljenim podjetjem.

Novice

Vsak neprijavljen uporabnik ob vstopu na spletno stran vidi vstopno stran - Novice, na kateri so objavljene novice o delovanju spletne strani ter povezave na Registracijo in Prijava. Do te strani lahko dostopajo vsi prijavljeni uporabniki.

O podjetju in kontakt

Tako registrirani kot neregistrirani uporabniki morajo imeti možnost gledanja zavihka O podjetju in Kontakt. Admin pa ima možnost še spreminjati to besedilo.

Pomoč

Na podstrani Pomoč morajo biti nazorno napisane razlage in navodila za uporabo te spletne strani.

Pogovor

Ta podzavihek ima na voljo le prijavljen uporabnik. Najprej je na njem razviden seznam vseh pogovorov določenega uporabnika, kjer so zapisana imena pogovorov (te lahko uporabniki poljubno spreminjajo). Če izberemo poljuben pogovor, se nam ta odpre (celotna zgodovina). Ob pogovoru so na voljo še hitre informacije in povezave na povpraševanja, podjetja ali investicije. To lahko spreminja vsak udeleženec pogovora.

1.2.2 Merila za ovrednotenje

Kvalitativna ocena naše rešitve nima smisla, saj je v okviru diplomske naloge izdelan le del aplikacije. Veliko bolj smiselno je rešitev ovrednotiti kvantitativno. Zato je aplikacija ovrednotena preko količine izdelane funkcionalnosti.

V okviru diplomske naloge smo, zaradi velikosti projekta, poskušali izdelati vsaj najbolj ključne funkcionalnosti aplikacije. Tako smo skušali izdelati vse funkcionalnosti uporabnikov: registracija uporabnikov, prijava in odjava, pregled uporabnikov, iskanje uporabnikov, spremembe profila in administracija uporabnikov. Poleg teh pa še vse funkcionalnosti podjetja: registracija podjetja, pregled podjetij, urejanje podjetja in prijava ter potrjevanje k podjetju. Izdelati smo želeli še funkcionalnosti povpraševanj: objava povpraševanj, pregled povpraševanj, iskanje povpraševanj, objava ponudbe, pregled ponudb in sklepanje pogodb. Poleg vseh teh funkcionalnosti, smo želeli izdelati še pogovor med uporabniki. Tako smo že v začetku izločili vse dele aplikacije, ki so povezane z investicijami in različnimi podstranmi, ki se bodo nahajale na strani.

Ocena strani je možna tudi v količini narejene varnosti na strani. V okviru diplomske naloge smo si tako zadali izdelati avtentikacijo preko žetonov (*tokens*) in preverjanje podatkov na strani strežnika.

1.3 Struktura naloge

Diplomska naloga je razdeljena na šest poglavij. V prvem poglavju je opisana željena končna funkcionalnost spletne aplikacije in merila s pomočjo katerih bomo skušali ovrednotiti delo. Sledi predstavitev drugih obstoječih rešitev, ki so vsaj nekoliko podobne naši. V tretjem poglavju je opisana tehnologija, ki je uporabljena pri implementaciji. Tej sledi poglavje, ki vsebuje procesni diagram in relacijsko entitetni model. Peto poglavje je namenjeno ovrednotenju funkcionalne rešitve in stopnje varnosti. V zadnjem poglavju pa je zapisan zaključek in načrti za nadaljnje delo.

Poglavje 2

Obstoječe rešitve

Obstaja veliko rešitev, ki so bolj ali manj podobne naši podjetniški ideji. Značilnost vsake izmed njih je, da je v določenih segmentih skoraj identična naši, v nekaterih pa se precej razlikuje. Izbrali smo tri že obstoječe rešitve, katerih cilji se vsaj delno prekrivajo z našimi. Te rešitve so Bolha, eBay in Europages.

2.1 Bolha

Bolha [2] je največja spletna tržnica v Sloveniji. Deluje tako, da neka fizična oseba ali podjetje brezplačno objavi neko ponudbo, ki je aktivna do enega meseca. V tem času lahko katerikoli član spletne strani kontaktira oglaševalca preko objavljenih podatkov in kupi kar želi. Cena je lahko fiksna že v začetku ali pa se o njej kupec in prodajalec dogovorita.

Bolha¹ in naša aplikacija sta si podobni v tem, da sta obedve nekakšni povezovalki med dvema stranema. Bolha skuša povezati oglaševalca in kupca, na naši spletni strani pa sta to povpraševalec in ponudnik. Nobena izmed teh strani pri tem ni dejansko vpletena v posel, kot je to v navadi pri klasičnih spletnih trgovinah.

Do neke mere je podoben tudi pregled in postopek iskanja. Na Bolhi so pod neko panogo v začetku vidni vsi izdelki, ko pa uporabimo iskalnik se njihov nabor zmanjša. Tudi pri naši aplikaciji je v začetku predstavljena lista povpraševanj ali

¹<http://www.bolha.com/>

podjetij, s filtracijo pa pridemo do tistih, ki jih želimo.

Poleg tega obedve omogočata tudi trgovanje s storitvami.

Strani pa si še zdaleč nista tako podobni, kot se morda zdi na prvi pogled. Najbolj očitna razlika je, da se Bolha osredotoča na poslovanje med podjetji in strankami ali pa celo samo med strankami (B2C in C2C). V naši podjetniški ideji pa smo osredotočeni predvsem na poslovanje med podjetji (B2B).

Sistema se razlikujeta tudi v tem, da Bolha skrbi samo za oglaševanje, nadaljna pogajanja in dejansko prodajo pa prepusti strankam s tem, da posreduje kontakt. V okviru naše aplikacije je poskrbljeno tudi za nadaljnje postopke kot sta zapis ponudbe in pogodbe.

2.2 eBay

eBay [14] je po svojem ustroju podoben Bolhi. Preko tega portala se trži stvari iz skoraj celega sveta. Trženje izgleda tako, da prodajalec najprej plača določen denarni znesek za njegovo objavo, se odloči za fiksno ali dražbeno ceno in nato objavi izdelek. Kupci lahko zatem kupijo objavljen izdelek po določeni ceni ali pa objavijo nek znesek v upanju, da bodo najboljši ponudnik na avkciji.

Ker je eBay² podobna aplikacija kot Bolha, so tudi nekatere podobnosti z našo aplikacijo enake. To sta vloga posrednika med kupcem in prodajalcem ter iskalnik.

Ena izmed stvari, ki jih Bolha nima, imata pa jo naša aplikacija in eBay, je možnost sprejemanja večih ponudb za isti izdelek. Bolha kakršnegakoli sprejemanja ponudb sploh ne pozna (posreduje samo kontakt), pri eBayu pa je to povsem mogoče in tudi uporabljeno - predvsem pri avkcijah. Tam namreč več ponudnikov poda svojo ponudbo, na koncu pa izdelek dobi tisti, ki je zanj pripravljen odšteti največ.

eBay vsebuje tudi spletno poslovanje. Kupec elektronsko plača prodajalcu, ta pa izdelek ponavadi pošlje po pošti ali pa posreduje kako drugače. Tudi naša spletna aplikacija poslovanje deloma izvede elektronsko in sicer vse do podpisa pogodbe.

eBay se od naše aplikacije, pa tudi od Bolhe, razlikuje po ciljni publiki. Tudi tu gre namreč za poslovanje med podjetji in končnimi uporabniki ali med samimi uporabniki izdelkov.

²<http://www.ebay.com/>

Razlika je tudi v tem, da eBay ne trži storitev, temveč le končne izdelke.

2.3 Europages

Spletna stran Europages³ vsebuje podatke o podjetjih po svetu, predvsem pa po Evropi (trenutno jih je registriranih okoli 2.6 milijona). Glavni namen spletne strani je povezovanje podjetij. To skušajo doseči tako, da vsako izmed njih objavi osnovne lastne podatke, panoge, na katerih sodelujejo in morebitne končne izdelke, ki jih izdelujejo. Uporabniki strani lahko v iskalnik vpišejo izdelek ali panogo zanimanja in tako na hiter način pridobijo seznam podjetij, ki razpolaga s tovrstnimi storitvami ali produkti.

Ciljni segment strani Europages je poslovanje med podjetji (B2B). Našemu delu je podobna v tem, da skuša povezati čim več podjetij preko podobnih parametrov. Omogoča namreč iskanje preko panog, imen izdelkov in imen podjetij.

Kot vse ostale opisane strani se tudi Europages ne spušča direktno v poslovanje, temveč prevzema vlogo posrednika kontaktov. Z našo stranjo ima skupno točko tudi v tem, da omogoča povezovanje preko storitev.

Omenjena stran je po vsebini in ciljni publiki zelo podobna naši, vseeno pa se strani v nekaj točkah tudi razlikujeta. Razhajata se v tem, da Europages ne podpira povpraševanj do te mere, kot jih mi. Tam namreč vedno dobiš podjetje, ki se ukvarja s to panogo, ne glede na to ali iščeš po produktih ali po storitvah. Na naši strani pa bo možno iskati po povpraševanjih samih. V tem segmentu je naša aplikacija veliko bolj podobna eBayu ali Bolhi kot pa strani Europages.

Naslednja točka razhajanja je v tem, da na Europages, podobno kot na Bolhi, uporabnika na svoji strani zadržijo le toliko časa, da pridobi ustrezne kontakte. S tem ne podpirajo nadaljnjega poslovanja kot sta objavljjanje ponudb ali sklepanje pogodb.

Stran Europages vsebuje bazo podjetij, ključne za iskanje in osnovne lastnosti podjetja, ne vsebuje pa ocen podjetij in števila sklenjenih poslov.

³<http://www.europages.com/>

2.4 Ostalo

Kot smo videli, je naša aplikacija nekakšen hibrid med dvema vrstama obstoječih rešitev. V objavah povpraševanj je podobna eBayu in Bolhi, v ciljni publiki in vsebini pa strani Europages.

Obstajajo še druge aplikacije, ki so podobne eBayu in Bolhi. To so na primer ebuyers.com⁴, Amazon.com⁵, shopping.yahoo.com⁶... Vse te strani se s funkcionalnostjo naše aplikacije prekrivajo v podobnih segmentih kot opisani aplikaciji, zato jim ne bomo posvečali posebne pozornosti.

Obstajajo tudi strani, ki so podobne Europages. Tudi te se med seboj razlikujejo le v podrobnostih, skupne točke z našo aplikacijo pa so pri vseh približno enake ali pa jih je še manj. Primera takšnih aplikacij sta Kompass.com⁷ in Yellowpages⁸.

⁴<http://www.ebuyer.com/>

⁵<http://www.amazon.com/>

⁶<https://shopping.yahoo.com/>

⁷<http://www.kompass.com/>

⁸<http://www.yellowpages.si/>

Poglavje 3

Uporabljene tehnologije

V tem poglavju so opisane tehnologije, ki so uporabljene v izdelavi spletne aplikacije.

3.1 PostgreSQL

PostgreSQL [20] je sistem za upravljanje objektno-relacijske podatkovne baze. Njegovi začetki segajo v leto 1977, ko se je začel projekt pod imenom Ingres. Leta 1996 se je preimenoval v PostgreSQL, saj je takrat postal odprtokoden. Danes je poznan kot najbolj napreden odprtokodni sistem za upravljanje s podatkovnimi bazami na svetu.

Ravno odprtokodnost je največja prednost PostgreSQL-a. To namreč pomeni, da lahko kdorkoli pridobi izvorno kodo in jo po željah spreminja. Posledično lahko dostopamo tudi do marsikaterih funkcionalnosti, ki so pri drugih ponudnikih plačljive. Tu gre predvsem za podatke o performančni statistiki.

PostgreSQL omogoča optimizacijo povpraševanj, dedovanja in seznamov (*array*).

Uporablja tehnologijo MVCC (*Multi-Version Concurrency Control*), s pomočjo katere se izogne zaklepanju. To dela tako, da sledi vsem izvedenim transakcijam za podatkovno bazo. Za uporabnike to pomeni, da ni več nepotrebnega čakanja na podatke v primerih, ko se ti spreminjajo.

V primeru prekinitve delovanja podatkovne baze PostgreSQL poskrbi za povrnitev podatkov preko zapisovanja vseh sprememb. Tako potem, ko je podatkovna

baza spet aktivna, najprej izvede ukaze, ki so bili podani že pred prekinitvijo.

Zaradi prožnosti PostgreSQL API-ja, je programom omogočena enostavna podpora temu sistemu. To je iz našega stališča ključnega pomena, pomeni namreč, da je uporaba PostgreSQL-a z jeziki kot so PHP, Ruby, C++, Java in predvsem Python preprosta.

3.2 Python

Python [15] je odprtokodni programski jezik. Razdeljen je na dve veji, Python 2 in Python 3. Delitev je nastala, ko so ustvarjalci jezika ugotovili, da določenih sprememb ne bo mogoče vključiti v starejšo verzijo (Python 2). Kot rešitev so izdelali novo verzijo jezika, ki ni skladna s starejšo. Danes so tako programerji postavljeni pred dilemo ali uporabljati starejšo verzijo, ki je dobro dokumentirana in je zanjo razvitih zelo veliko knjižnjic, ali pa slediti najnovejšim smernicam programskega jezika.

Razlogov za tako množično uporabo tega jezika (uporabljalo naj bi ga okoli milijon programerjev) je več. Eden izmed glavnih je verjetno kvaliteta programske opreme. Za razliko od večine drugih programskih jezikov je Python narejen tako, da je njegova sintaksa enostavnejša in posledično lažje berljiva. To pomeni, da so iskanje napak, prebiranje tuje kode in optimizacija rešitev precej enostavnejši kot drugje.

Python poveča tudi produktivnost razvijalca. To je posledica tega, da je koda v tem jeziku, po besedah Marka Lutza, ponavadi dolga od ene petine do ene tretjine kode zapisane v jeziku Java ali C++.

Jezik dobro podpira integracijo komponent. Konkretno lahko uporablja knjižnjice jezikov C in C++ ter vgradi določene komponente iz Java ali C++.

Dobra lastnost jezika je odlična podpora knjižnjicam, saj že sama standardna knjižnjica vsebuje veliko množico predhodno izdelane funkcionalnosti. Še večjo dodano vrednost pa predstavljajo knjižnjice, ki jih niso razvili razvijalci Pythona pač pa njihovi uporabniki. Knjižnjice so namenjene raznovrstnim stvarem, kot so numerično programiranje, razvoj računalniških iger in konstrukcija spletnih strani.

3.2.1 Django

Django [6,11] je spletno ogrodje (*framework*), ki je napisano v jeziku Python. Razvit je bil leta 2003 s strani dveh programerjev, ki sta se ukvarjala z razvojem večih strani časopisnih hiš. Ugotovila sta, da se velik del kode pri vsakem novem projektu ponovi, zato sta te dele združila in tako ustvarila Django.

Spletno ogrodje je koda, ki tvori programsko infrastrukturo spletne aplikacije. Njen glavni namen je pomagati programerju pri strukturiranju kode in programiranju kode, ki se jo da vzdrževati.

Za enostavnejše programiranje je kodo potrebno ustrezno strukturirati. Med spletnimi aplikacijami se je tako pojavila strukturna shema MVC (*Model-View-Controller*) [5], ki jo uporablja tudi Django. MVC strukturno razdeli aplikacijo na tri dele: model(*model*), pogled(*view*) in nadzor(*controller*). Model vsebuje kodo, ki je povezana z definiranjem in dostopanjem do podatkov, nadzor naj bi vseboval logiko aplikacije, pogled pa dejanski prikaz slike. Prednost takšne delitve programske kode je v tem, da so te komponente med sabo povezane le ohlapno. Posledično velikokrat lahko spremenimo samo kodo v posamezni komponenti, brez vpliva na kodo znotraj drugih komponent.

3.2.2 Django REST ogrodje

Django REST ogrodje [3] je kot Django eden izmed ogrodij razvitih za Python. Izpeljan je iz Djanga in spremenjen do te mere, da ustreza REST-u.

REST [7] (*Representational State Transfer*) je protokol za prenos podatkov med odjemalcem in strežnikom, ki je sestavljen iz večih omejitev. Z omejevanjem so dosežene določene lastnosti, ki programerju poenostavljajo delo. Konkretno s tem dosežemo povečljivost, preprostost vmesnikov, spremenljivost arhitekture, vidljivost interakcij znotraj aplikacije, prenosljivost na različna okolja in boljšo zanesljivost delovanja.

Django REST ogrodje še vedno sledi shemi MVC. Če ga primerjamo z Djangom ugotovimo, da imata enak model, nadzora se nekoliko razlikujeta, saj oddajata in sprejemata različne tipe podatkov, pogled pa je skoraj povsem spremenjen. Django v tej fazi uporablja nek svoj jezik - *Django template language* - ki je mešanica jezikov

HTML in Pythona, Django REST ogrodje pa že znotraj nadzora poskrbi, da se podatki spremenijo tako, da jih lahko uporabijo skoraj vsi jeziki predstavitvenega sloja (v standarden format JSON ali XML). S tem prepusti implementacijo pogleda drugim, poljubnim programerskim okoljem, ki znajo uporabljati omenjena formata.

3.3 JavaScript

Jezik JavaScript [8,18] je bil razvit leta 1995. Razvilo ga je podjetje Netscape v želji, da bi s pomočjo internetnih brskalnikov in strežnikov ustvarili prenosni operacijski sistem. JavaScript je tako nastal ob spoznanju, da sama HTML koda ni zadosti za razvoj aplikacij na brskalniku.

JavaScriptu bi lahko rekli jezik spleta. Danes ga uporabljajo skoraj vse spletne strani, vsi moderni spletni brskalniki (na računalnikih, pametnih telefonih, tablicah...). Tako je najbolj razširjen jezik na svetu. Praktično vsi spletni razvijalci morajo poznati njegove osnove.

S programerskega jezika bi lahko za JavaScript rekli, da je visoko-nivojski, dinamični, netipiziran in interpretiran jezik. Podpira tako objektno usmerjeno programiranje kot tudi funkcijsko. Po sintaksi je precej podoben jeziku Java, vendar sta to dva povsem različna jezika, ki se razlikujeta že v samih osnovah.

JavaScript sam po sebi definira samo delo s tekstom, datumi, polji in regularnimi izrazi, ne vsebuje pa zelo pomembne funkcionalnosti kot so vhodi in izhodi. Za jezik bi tako lahko rekli, da je vgrajen v neko okolje, ki poskrbi za ostale operacije, ki so ključne za delovanje. Največkrat so to spletni brskalniki, kar pa ni nujno.

3.3.1 AJAX

JavaScript je bil v začetku prisoten na spletnih straneh zato, da so te postale dinamične. Ta pristop je uporabljen še danes, pojavili pa so se še drugi, prav tako zelo uporabni. Tako je Microsoft sprožil pravo revolucijo, ko je omogočil AJAX [9] klice.

AJAX-u (*Asynchronous JavaScript and XML*) bi lahko rekli, da je nekakšen strežnik na strani brskalnika. Arhitekturno je predstavljen kot plast, ki omogoča

izdelavo HTML in CSS kode znotraj brskalnika. S spletnim strežnikom se lahko pogovarja preko asinhronih klicev in tako prenaša samo del kode. Ta optimizacija je pomembna predvsem takrat, ko se spremeni samo del spletne strani, ostalo pa ostane takšno, kot je bilo.

3.3.2 JSON

JSON [4] (*JavaScript Object Notation*) je tekstoven format namenjen izmenjavi strukturiranih podatkov. V svetu spletnega programiranja je uporaben predvsem pri prenosu podatkov s strežnika in na strežnik.

Z njim lahko predstavimo tri osnovne tipe - nize črk, števila in tip boolean. Podpira pa tudi predstavitev dveh strukturiranih tipov - objektov in seznamov (*arrays*). Seznami so urejena zbirka podatkov, objekt pa je predstavljen kot neurejena zbirka podatkov, kjer je ime vedno predstavljeno kot niz znakov, vrednost pa je lahko katerikoli izmed primitivnih ali strukturiranih tipov.

Poseben format, ki izhaja iz JSON-a je tudi JWT [12] (*JSON Web Token*). Od navadnega objekta tipa JSON se razlikuje v tem, da je prostorsko omejen, zakodiran in digitalno podpisan. Uporabljen je pri avtorizaciji znotraj HTTP glave ali pri povpraševanju preko URL parametrov.

3.3.3 AngularJS

AngularJS [10] je strukturno ogrodje (*framework*) za izgradnjo dinamičnih spletnih aplikacij. Omogoča, da se statična HTML koda približa dinamični. Za programerja to pomeni, da mu ni potrebno pisati toliko ponavljajočega HTML-ja kot sicer. Zelo pomembna lastnost ogrodja je podpora eno-stranske aplikacije (*single-page application*).

AngularJS predstavlja rešitev na strani odjemalca (*client-side solution*). To stori tako, da nadzoruje DOM (*Document Object Model*) in prenos podatkov s strežnika. Njegov cilj je, da celotno JavaScript kodo strukturira tako, da omogoča izgradnjo CRUD (*create-read-update-delete*) aplikacije. Poleg tega omogoča tudi testiranje z unit-testi, end-to-end testi ipd. Ker je samo po sebi ogrodje, vsebuje še začetno kodo, iz katere lahko zrastejo večje aplikacije.

Njegove glavne značilnosti so povezovanje podatkov med seboj, nadzor nad DOM-om za skrivanje ali ponavljanje delov, podpora obrazcem in njihovim validacijam, pripenjanje kode iz ozadja na DOM elemente in grupiranje HTML-ja v ponovno uporabljene dele.

3.4 Spletno programiranje

3.4.1 HTML

HTML [16] (*HyperText Markup Language*) je označevalni jezik, namenjen izdelavi spletnih strani. Sestavljen je iz tako imenovanih značk, ki razdelijo celoten dokument na več delov. Spletni brskalniki so zmožni prebirati tako zapisane podatke in iz njih izdelati vizualno sliko. To storijo tako, da značke uporabijo za določitev načina prikaza podatkov, vsebino pa izpišejo takšno kakršna je.

3.4.2 CSS

CSS [19] (*Cascading Style Sheets*) pove brskalniku kako prikazati posamezne HTML elemente. Z drugimi besedami, skrbi za prezentacijo spletne strani. Omogoča spreminjanje barv, pisav in postavitve strani (*layout*). Pomembna lastnost je tudi ta, da lahko spreminja izgled spletne strani glede na velikost zaslona. Sicer je CSS podobno kot HTML označevalni jezik. Ponavadi je zaradi preglednosti ločen od HTML dokumenta.

3.4.3 Bootstrap

Bootstrap [17] je ogrodje (*framework*), ki je namenjen preprostejšemu in hitrejšemu spletnemu programiranju. To je možno zato, ker vsebuje množico predlog, sestavljenih iz HTML-ja in CSS-ja. Tako imajo uporabniki bootstrapa že predhodno pripravljen izgled gumbov, tabel, zavihkov...

Glavna prednost Bootstrapa je ta, da uporabniku prihrani veliko časa, saj se mu ni potrebno ukvarjati z definicijo posameznih elementov. Dodana vrednost pa je tudi

ta, da je z njegovo uporabo stran odzivna na velikost prikaza slike in je izgled konsistenten. Poleg tega je relativno preprost za uporabo in združljiv z najrazličnejšimi brskalniki. Tudi Bootstrap je odprtokoden.

Poglavje 4

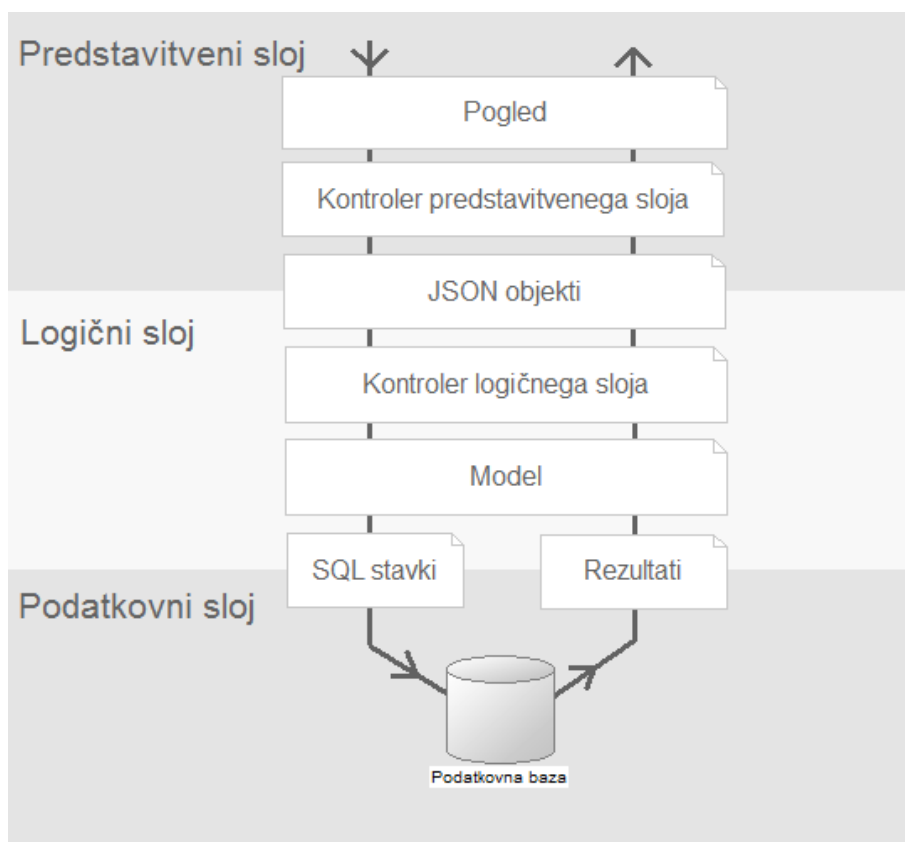
Opis rešitve

4.1 Arhitektura aplikacije

V aplikaciji je uporabljena 3-slojna arhitektura [13] (Slika 4.1). To je arhitektura, ki je v osnovi sestavljena iz predstavitvenega, logičnega in podatkovnega sloja.

Prednosti tovrstne razdelitve so v tem, da je vsak izmed slojev robusten in neodvisen. Posledično lahko vsak izmed njih uporablja svojo strojno opremo, s čimer dosežemo boljše preformance. Neodvisnost slojev pomeni tudi to, da v primeru povsem spremenjenega nivoja, druga dva lahko останeta enaka in bo rešitev še vedno delovala. Konkretno v primeru sprememb na predstavitvenem sloju nam logičnega in podatkovnega nivoja ni potrebno spreminjati. Prednost tovrstne delitve je tudi neodvisnost programskih tehnologij po nivojih. To pomeni, da lahko spremenimo uporabljeno tehnologijo na določenem nivoju brez oziroma z minimalnimi spremembami na drugih nivojih.

Naša aplikacija na predstavitvenem sloju uporablja klasične tehnologije za prikaz na spletu, kot so HTML, CSS in JavaScript. V okviru JavaScripta velja izpostaviti AngularJS, saj ta vpliva na nekoliko specifično sestavo tega nivoja. Logični sloj je napisan v Pythonu, natančneje z uporabo ogrodja Django REST Framework. Na podatkovnem sloju pa je uporabljena relacijska podatkovna baza PostgreSQL.



Slika 4.1: Shema arhitekture naše aplikacije.

4.1.1 Predstavitveni sloj

Predstavitveni sloj je sestavljen iz osnovnih tehnologij, za izgradnjo spletnih strani: HTML, CSS in JavaScript. Kot pa sem opisal že v uvodu ima na tem sloju posebno vlogo tudi AngularJS. Ta namreč vpliva na to, da se predstavitveni sloj deli po sistemu MVC [1].

Pogled je napisan v jeziku HTML, za njegovo obliko pa poskrbi datoteka formata CSS. AngularJS na tem mestu prinaša spremembe tako, da razširi slovar jezika HTML z mehanizmom direktiv (*directives*). Te HTML-ju dodajo možnost izvajanja nalog programskih jezikov kot so iteracija čez seznam podatkov, evalvacija pogojnih izrazov in podobno.

Nadzor je zapisan v JavaScriptu in predstavlja zapis logike, ki je potrebna, da podatke iz modelov preslikamo v podatke za prikaz. Ta del lahko zaradi preglednosti tudi razširimo s tem, da v arhitekturo dodamo tako imenovane storitve (*services*). V tem primeru se nadzori in storitve med sabo delijo tako, da so nadzori zadolženi predvsem za pripravo podatkov za izpis, storitve pa naj bi vsebovale večino logike. Za povezavo med nadzorom in pogledom sicer skrbi poseben objekt, do katerega lahko dostopata obe strani (*scope*).

V tovrstno arhitekturo spada tudi model. Zapisan je kot JavaScript objekt - POJO (*Plain-Old-JavaScript-Object*). Od ostalih delov arhitekture se razlikuje v tem, da ti modeli fizično ne obstajajo. Podatke same namreč v tem primeru v predstavitevni sloj dobimo preko REST klicev na logični sloj. Tako komunikacije med logičnim in predstavitvenem nivojem potekajo v obliki objektov tipa JSON.

4.1.2 Logični sloj

Logični sloj je zapisan v jeziku Python. Pri tem je uporabljeno ogrodje Django REST Framework, ki izhaja iz Djanga. Arhitekturno gledano je Django sestavljen po sistemu MVC, za Django REST Framework (Poglavje 3.2.2) pa bi lahko rekli, da je sestavljen le iz modela in nadzora.

Pogleda v okviru Django REST Frameworka ni. Ta namreč predvideva, da se pogled ne bo zgeneriral na strežniku, pač pa pri odjemalcu.

Nadzor je znotraj tega ogrodja sestavljen iz dveh obveznih delov. To sta pogled (*view*) in serializator (*serializer*). Znotraj pogleda je zapisana vsa logika delovanja, torej avtentikacija, validacija in tako dalje. Glavni namen serializatorja pa je pretvorba objektov iz Pythona v JSON objekte, ki se dejansko pošiljajo predstavitvenemu sloju.

Model znotraj Django REST Frameworka je enak modelu iz Djanga. Sicer model v tem primeru pomeni preslikavo podatkov iz podatkovne baze na objekte v jeziku Python. Tako vidimo, da so dostopi do podatkovne baze možni le preko modelov, ki so lahko klicani iz pogleda znotraj Django REST Frameworka.

4.1.3 Podatkovni sloj

Podatki sami so shranjeni na podatkovnem sloju. V naši aplikaciji je uporabljena relacijska podatkovna baza PostgreSQL. Za poizvedovanje po njej se uporablja jezik SQL. Programerjem ob uporabi Djanga ali Django REST Frameworka ni potrebno skrbeti za tovrstne klice, saj jih namesto njih naredi ogrodje. To na podlagi modelov kreira tabele in nato vstavlja, bere, spreminja ali briše podatke znotraj teh tabel.

4.2 Procesni diagram

Na tem mestu bomo predstavili procese naše aplikacije, ki smo jih izdelali. V osnovi so razdeljeni na procese, ki se nanašajo na uporabnike, podjetje in poslovanje. Za takšno delitev smo se odločili zaradi preglednosti in navezovanja procesov na skupno točko.

4.2.1 Uporabnik

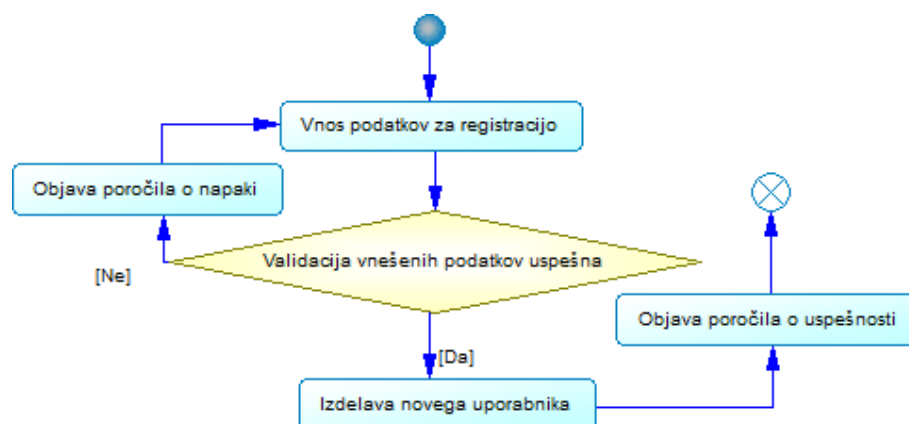
Procesi uporabnika so edinstveni v tem, da se navezujejo na uporabnika samega, torej na njegovo ustvarjenje, upravljanje, spreminjanje ipd.

Procesi uporabnika so:

- Registracija uporabnika
- Spreminjanje profilnih podatkov
- Prijava

Registracija uporabnika

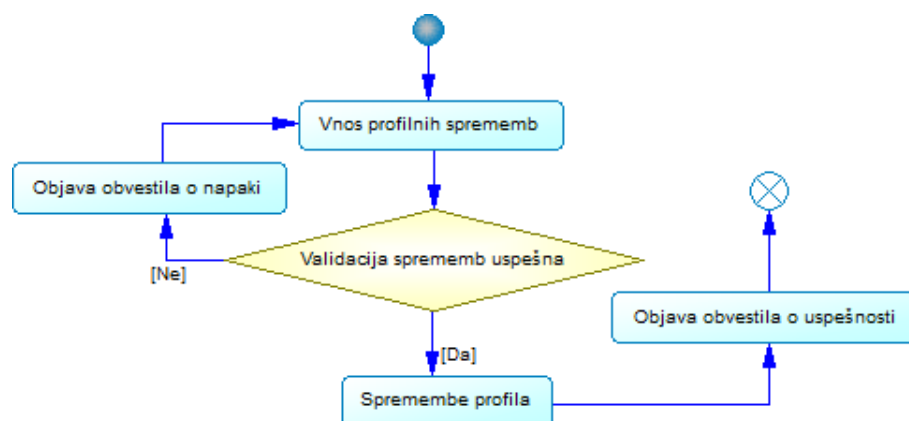
Vsakdo se lahko registrira v sistem (Slika 4.2). Pri tem dobi pooblastilo navadnega uporabnika.



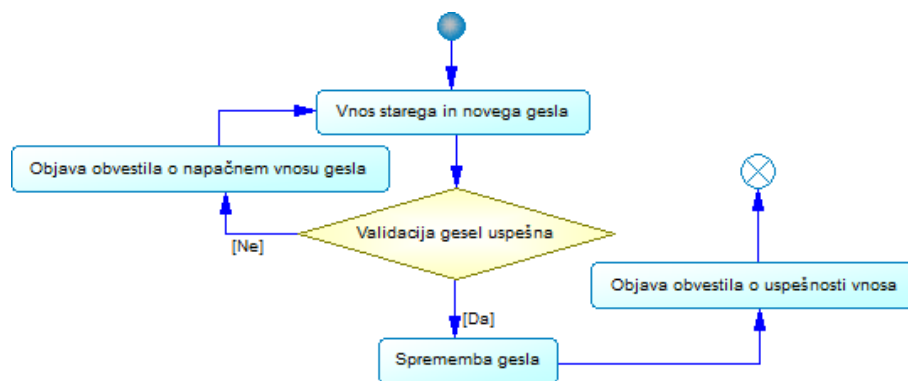
Slika 4.2: Aktivnostni diagram poteka registracije.

Spreminjanje profilnih podatkov

Vsak uporabnik ima možnost spreminjanja svojih osebnih podatkov (Slika 4.3). V primeru, da je pozabil geslo, ima tudi možnost njegove ponastavitve (Slika 4.4), ob kateri je potrebno ponovno vnesti tudi geslo. Podatke o uporabniku lahko spreminja tudi administrator. Pri tem lahko spremeni tudi vlogo posameznega uporabnika (Slika 4.5).



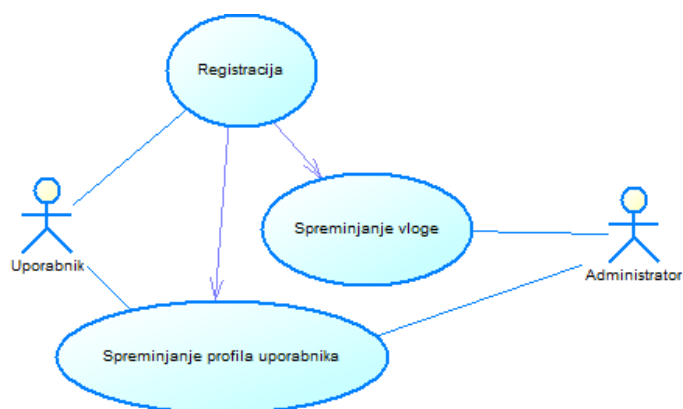
Slika 4.3: Aktivnostni diagram spreminjanja profila.



Slika 4.4: Aktivnostni diagram spreminjanja gesla.

Prijava

Vsi registrirani uporabniki se lahko vpišejo v sistem s pomočjo uporabniškega imena in gesla (Slika 4.6).



Slika 4.5: Use case diagram spreminjanja profila in vloge.

4.2.2 Podjetje

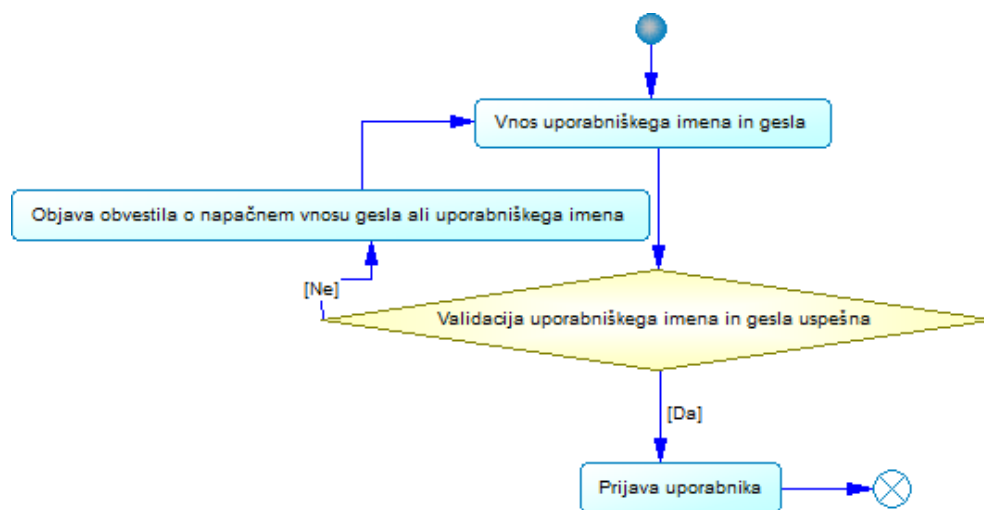
Procesi podjetja imajo skupno točko v tem, da se vsi navezuje na podjetja.

Procesi podjetja so:

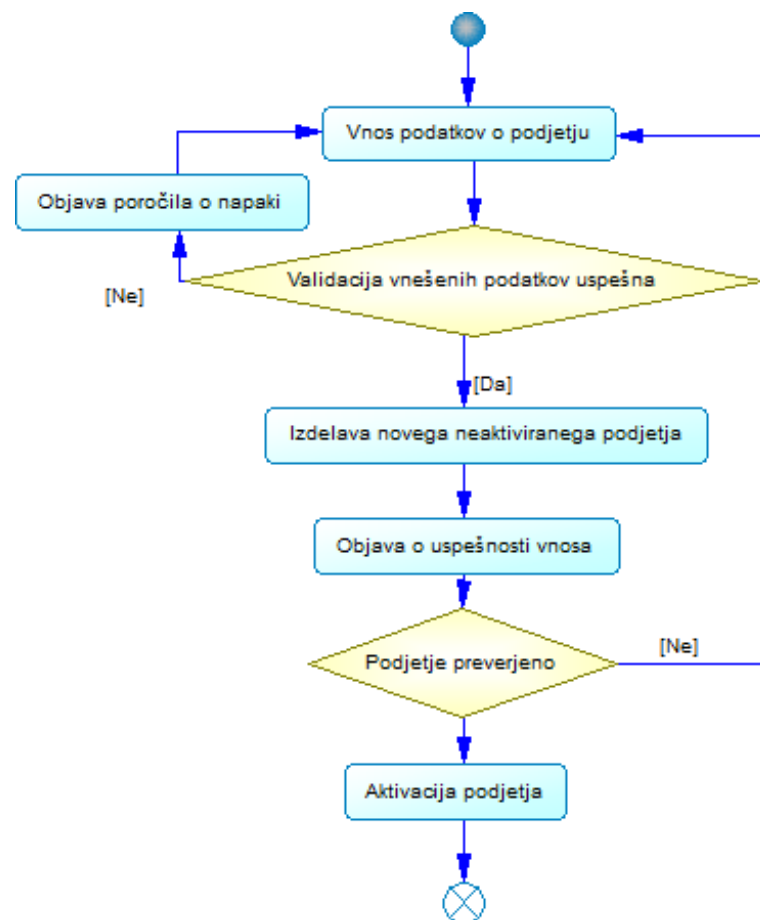
- Registracija podjetja
- Popravljanje podatkov
- Prijava k podjetju
- Pogovor med uporabniki
- Pregledovanje podjetij

Registracija podjetja

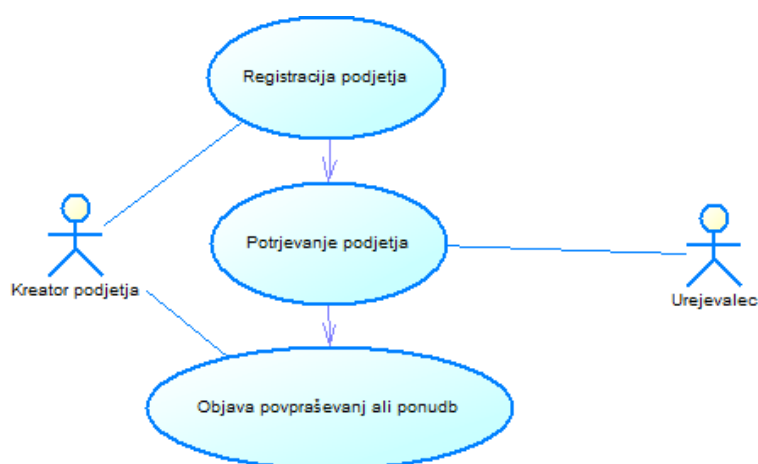
V tem procesu navaden uporabnik najprej izpolni osnovni obrazec za tvorjenje predstavitve podjetja. Po uspešnem vnosu podatkov se podjetje tvori in njegovemu ustvarjalcu avtomatsko dodeli vse pravice. Sledi čakanje na urejevalca, ki preveri dejansko pristojnost uporabnika naše strani v registriranem podjetju. Če je podjetje pristojno, ga potrdi (Slika 4.7). Če podjetje ni potrjeno, ne more objavljati povpraševanj in ponudb ter ni vidno na seznamu podjetij (Slika 4.8).



Slika 4.6: Aktivnostni diagram prijave.



Slika 4.7: Aktivnostni diagram registracije podjetja.



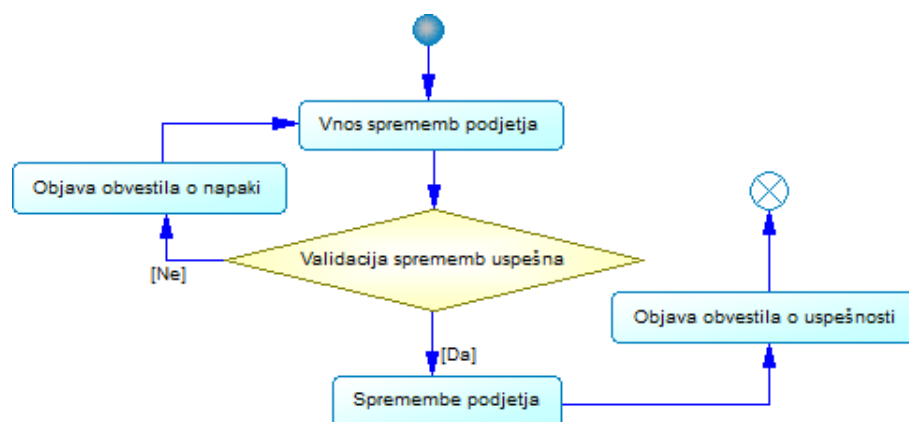
Slika 4.8: Use case diagram registracije podjetja.

Popravljanje podatkov

Vsi uporabniki, ki so za to pooblaščen, imajo možnost spreminjanja katerihkoli podatkov o njihovem podjetju, razen imena. Ime lahko spremenijo le administratorji (Slika 4.9).

Prijava k podjetju

V primeru, da v nekem podjetju to spletno stran uporablja več delavcev, lahko ti, potem ko najdejo profil podjetja, zaprosijo za prijavo (Slika 4.10). Po zaprositvi jih za to pooblaščen uporabnik lahko potrdi in jim določi pooblastila znotraj podjetja (Slika 4.11).



Slika 4.9: Aktivnostni diagram sprememb podjetja.

Pregledovanje podjetij

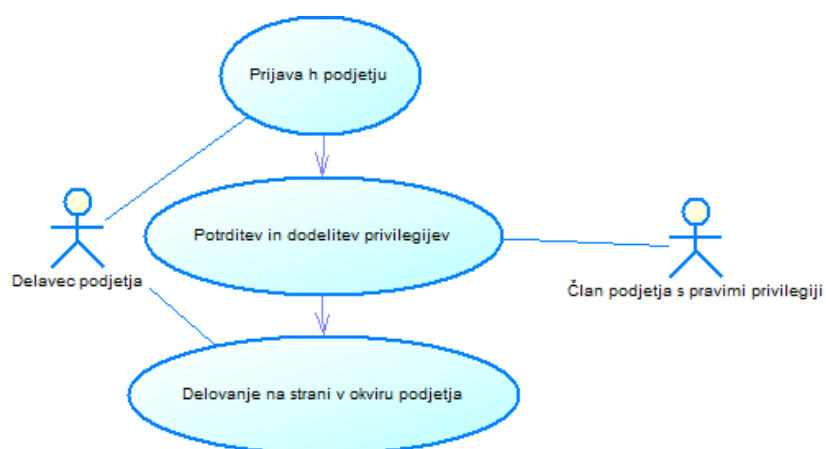
Vsak član strani lahko vidi seznam vseh podjetij na strani. Po želji uporabi iskalnik, tako da so vidna le tista, ki ustrezajo iskalnim parametrom (Slika 4.12). Lahko izbere podjetje, nakar se mu prikažejo njegove podrobnosti.

4.2.3 Poslovanje

Pod poslovanje uvrščamo vse tiste procese, ki se navezujejo na dejansko poslovanje (Slika 4.13).

Procesi podjetja so:

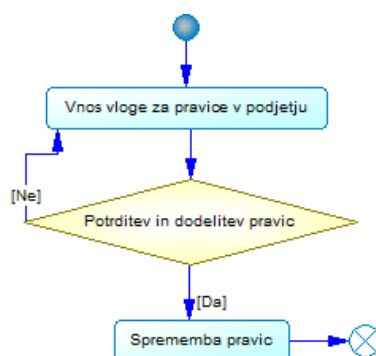
- Povpraševaje po poslu
- Iskanje po povpraševanjih posla
- Ponudba na povpraševanje posla
- Pregled po ponudbah posla
- Sklepanje pogodbe



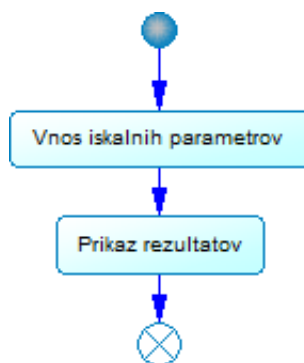
Slika 4.10: Use case diagram spreminjanja pravic uporabnikov v okviru podjetja.

Povpraševaje po poslu

Pri tem procesu lahko uporabnik, ki je član nekega podjetja in ima za to pooblastila, objavi povpraševanje po željenem delu (Slika 4.14).



Slika 4.11: Aktivnostni diagram spreminjanja pravic uporabnikov v okviru podjetja.



Slika 4.12: Aktivnostni diagram pregledovanja podjetij.

Iskanje po povpraševanjih posla

Vsi registrirani uporabniki lahko pregledujejo povpraševanja in ob morebitnih nejasnostih kontaktirajo osebe, ki so odgovorne zanje. Pri pregledovanju imajo tudi možnost filtrov, s pomočjo katerih lahko hitreje najdejo željeno povpraševanje (Slika 4.15).

Ponudba na povpraševanje posla

Registrirani uporabniki, ki spadajo pod neko podjetje, lahko pošljejo uradno ponudbo na povpraševanje (Slika 4.16). Ti uporabniki morajo znotraj podjetja imeti zadostna pooblastila za to.

Pregled po ponudbah posla

Vsak za to pooblaščen član podjetja, ki je objavilo neko povpraševanje, lahko pregleduje vse uradne ponudbe, ki so prišle za neko delo. Če želi, sprejme neko ponudbo in nadaljuje v fazo sklepanja pogodbe.

Urejanje pogodbe

Tako povpraševalec kot ponudnik lahko urejata isto besedilo - pogodbo - ki je v začetku prekopirana ponudba. Ko sta obe strani zadovoljni, to potrdita. Po tem ni več možno spreminjati dokumenta, oba pa lahko ocenita partnerja pri delu (Slika 4.17).

4.3 Entitetno relacijski model

Za lažjo predstavo delovanja aplikacije so na kratko opisane glavne entitete in relacije med njimi. Najpomembnejše entitete so:

- Uporabnik
- Podjetje
- Povpraševanje
- Ponudba

4.3.1 Entitete

Uporabnik

Entiteta Uporabnik je pomembna, ker vsebuje podatke o določenem uporabniku. Glavni atributi so tako uporabniško ime, e-naslov, geslo in ime vloge, ki jo ima izbran uporabnik (administrator, urejevalec ali navaden uporabnik).

Podjetje

Entiteta Podjetje vsebuje vse podatke o podjetju. Njegovi najpomembnejši atributi so ime, panoga, opis, naslov, število opravljenih del, ocena podjetja in atribut, ki pove, ali je bilo podjetje potrjeno s strani urejevalca ali ne.

Povpraševanje

V okviru entitete Povpraševanje so zajeti podatki o specifičnem povpraševanju. Entiteta tako vsebuje ime, panogo, podrobnosti, datum aktivnosti in podjetje, ki je objavilo povpraševanje.

Ponudba

Entiteta ponudba zajema vse podatke, ki so povezani s ponudbo in kasneje pogodbo. Vsebuje povpraševanje na katero se navezuje, podjetje, ki objavlja ponudbo, podrobnosti pogodbe, atribut, iz katerega izvemo, če je bila ponudba sprejeta tako, da ji

lahko rečemo pogodba, in dva atributa, ki povesta, če je bila pogodba potrjena s strani povpraševalca in ponudnika.

4.3.2 Relacije

Uporabnik - Podjetje

Entiteti Uporabnik in Podjetje, sta med sabo povezani s povezavo več na več (*many-to-many*). Uporabnik namreč lahko dela pri večih podjetjih, tudi podjetja pa imajo lahko več uporabnikov (Slika 4.18).

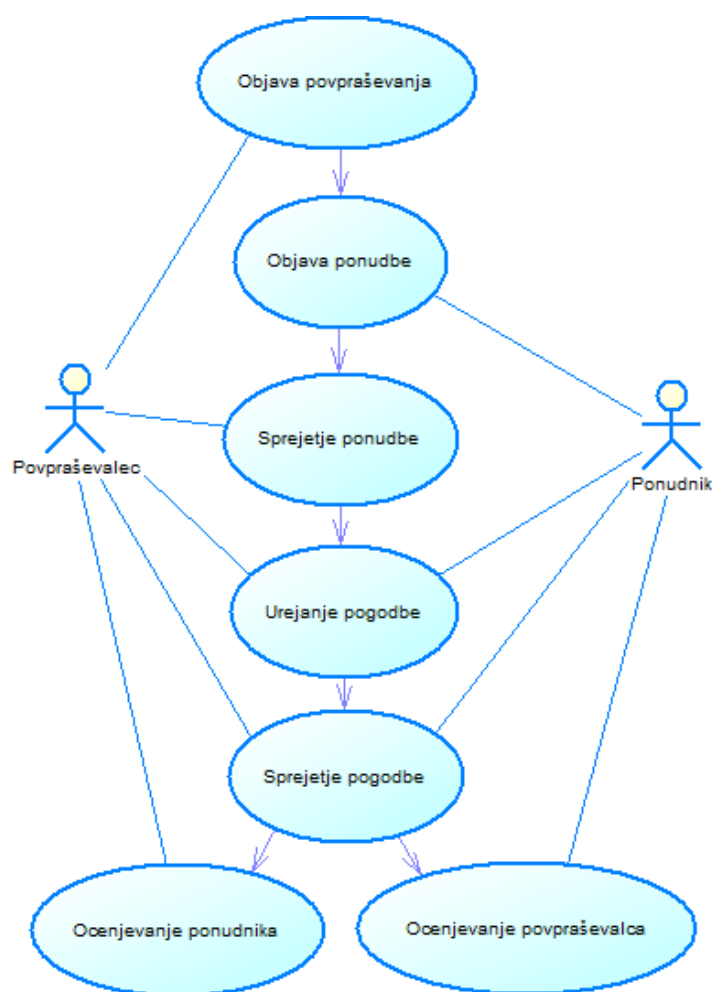
Seveda povezava več na več v relacijskih modelih ni mogoča, zato smo dodali novo entiteto - Pooblastila. Ta entiteta poleg tujih ključev uporabnika in podjetja namreč vsebuje tudi podrobna pooblastila določenega uporabnika znotraj podjetja (Sliki 4.19, 4.20 in 4.21).

Podjetje - Povpraševanje

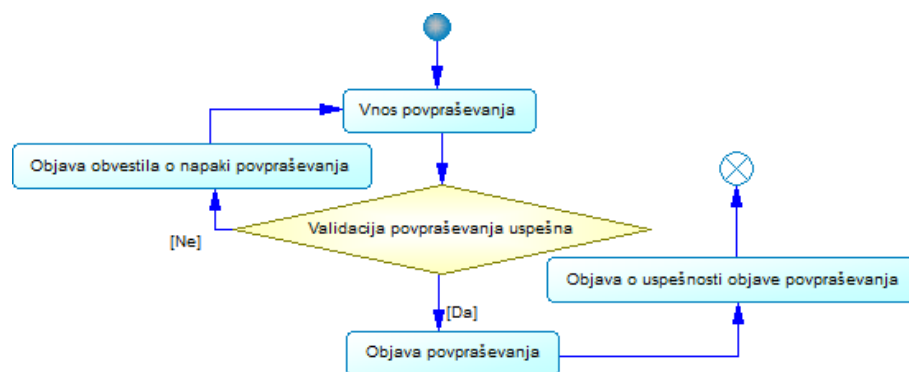
Določeno podjetje lahko objavi več povpraševanj, vsako povpraševanje pa seveda naredi samo eno podjetje (Slika 4.22). Tako je ta relacija triviala, ena na več (*one-to-many*).

Podjetje - Ponudba - Povpraševanje

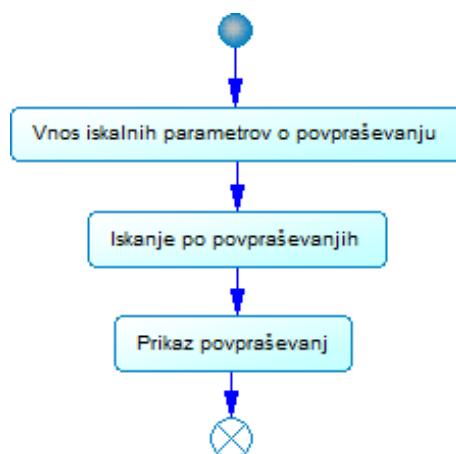
Ponudba je odvisna tako od podjetja, ki jo ustvari, kot tudi od povpraševanja, na katerega se navezuje. Kar se tiče relacij med temi entitetami lahko neko podjetje ustvari več ponudb (Slika 4.23), tudi na neko povpraševanje pa lahko pride več ponudb (Slika 4.24).



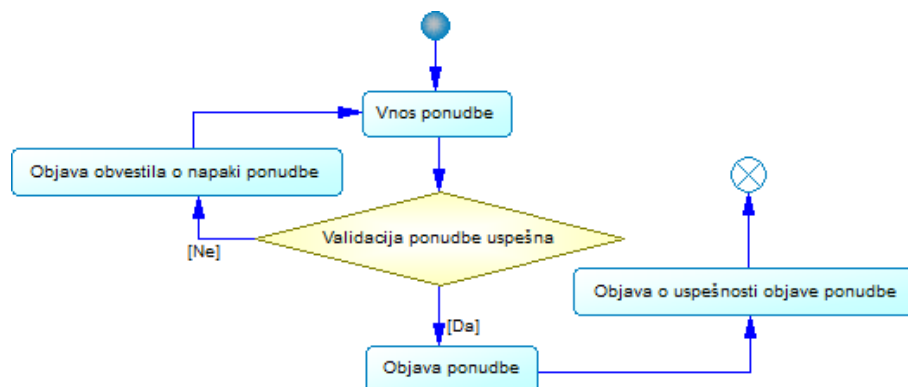
Slika 4.13: Use case diagram poslovanja.



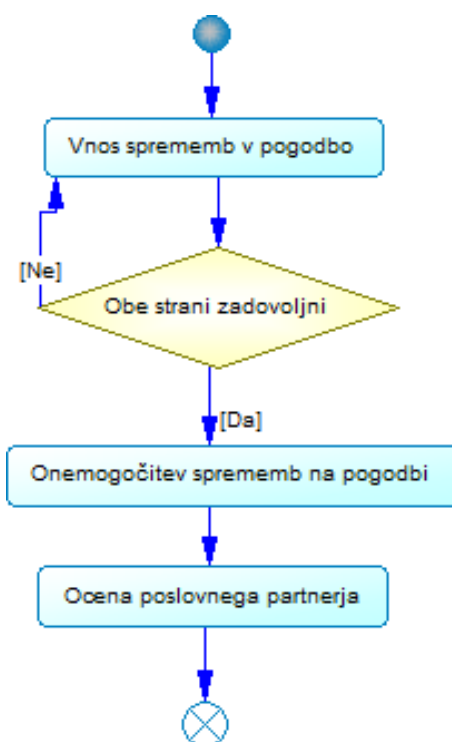
Slika 4.14: Aktivnostni diagram vnosa povpraševanja.



Slika 4.15: Aktivnostni diagram iskanja povpraševanja.



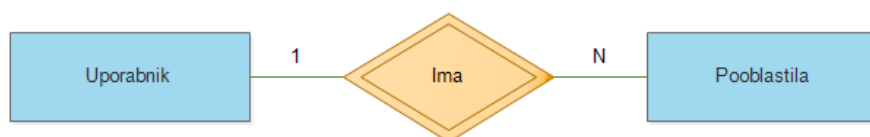
Slika 4.16: Aktivnostni diagram vnosa ponudbe.



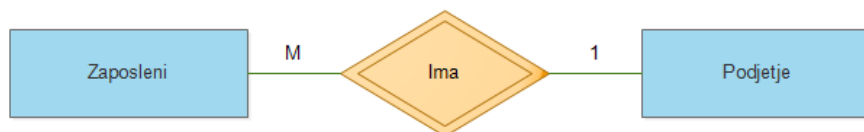
Slika 4.17: Aktivnostni diagram urejanja pogodbe.



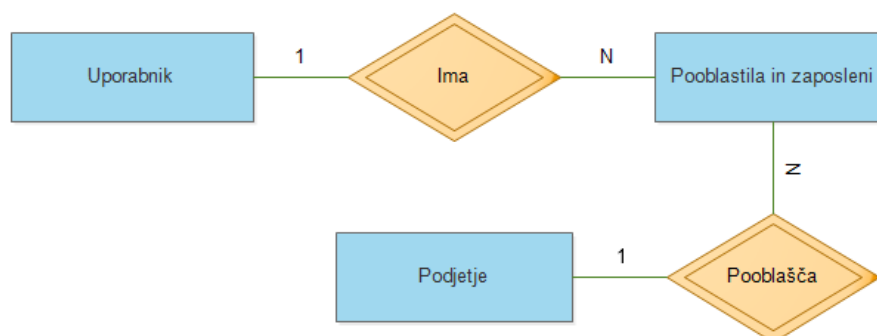
Slika 4.18: Entitetno relacijski diagram relacije med uporabnikom in podjetjem.



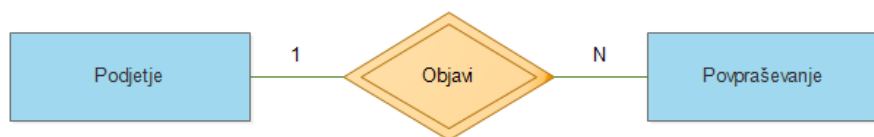
Slika 4.19: Entitetno relacijski diagram relacije med uporabnikom in pooblastilom.



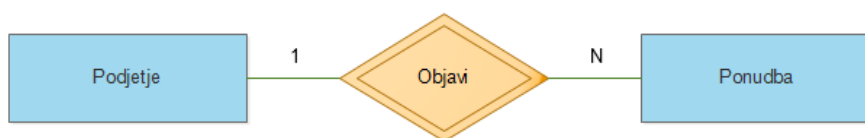
Slika 4.20: Entitetno relacijski diagram relacije med zaposlenim in podjetjem.



Slika 4.21: Entitetno relacijski diagram relacije med uporabnikom, pooblastili in podjetjem.



Slika 4.22: Entitetno relacijski diagram relacije med podjetjem in povpraševanjem.



Slika 4.23: Entitetno relacijski diagram relacije med podjetjem in ponudbo.



Slika 4.24: Entitetno relacijski diagram relacije med ponudbo in povpraševanjem.

Poglavje 5

Ovrednotenje rešitve

V tem poglavju so opisani implementirani deli sistema in varnostni vidik aplikacije.

5.1 Funkcionalnost

Celotni projekt zajema veliko funkcionalnosti. Zato v okviru diplomske naloge ni izdelana vsa, temveč le toliko, da smo vzpostavili prototipni del aplikacije. Napisan je tisti del kode, zaradi katerega se naša aplikacija najbolj razlikuje od drugih.

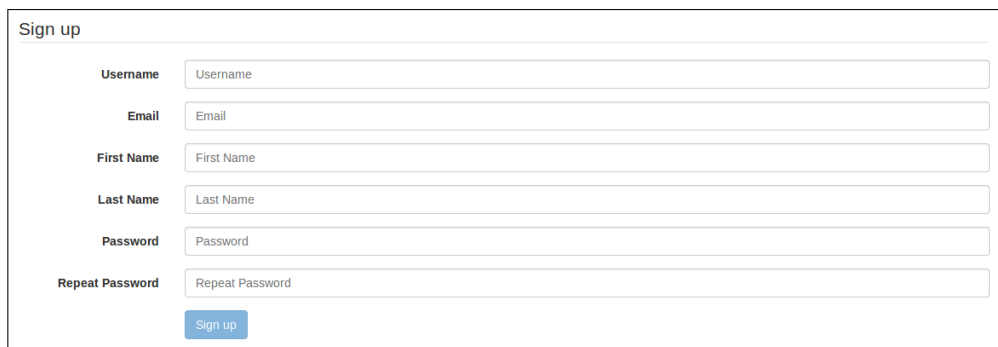
5.1.1 Dostop do aplikacije

Spletna stran je dostopna na naslovu: `http://poslovnaideja.fri.uni-lj.si/`. Za celovito testiranje strani je potrebno poznati uporabniško ime in geslo administratorja. Na omenjenem naslovu je administratorsko uporabniško ime “admin”, geslo pa prav tako “admin”.

5.1.2 Slikovna predstavitev funkcionalnosti

Za boljšo predstavo kakšen del je že implementiran se nam zdi najbolj smiselno, da naštejemo uporabniške zgodbe, ki smo jih izdelali iz poglavja 1.2.1 in preko slik zaslona povzamemo funkcionalnost:

- Registracija uporabnika (Slika 5.1)



A sign up form with the title "Sign up" at the top left. It contains six input fields, each with a label to its left: "Username", "Email", "First Name", "Last Name", "Password", and "Repeat Password". Each input field has a light gray placeholder text matching its label. Below the "Repeat Password" field is a blue button with the text "Sign up" in white.

Slika 5.1: Obrazec za registracijo uporabnika.

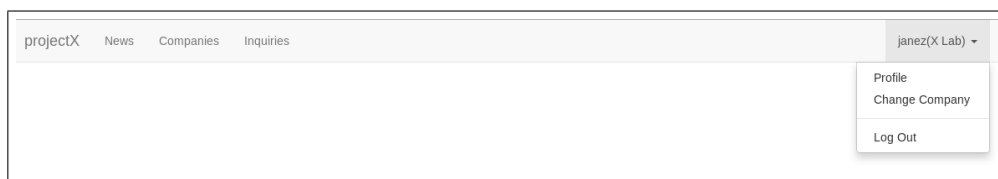
- Prijava (Slika 5.2)



A sign in form with the title "Sign in" at the top left. It contains two input fields, each with a label to its left: "Username" and "Password". Each input field has a light gray placeholder text matching its label. Below the "Password" field is a blue button with the text "Sign in" in white.

Slika 5.2: Obrazec za prijavo uporabnika.


- Odjava (Slika 5.3)



A user menu interface. At the top, there is a horizontal navigation bar with links: "projectX", "News", "Companies", and "Inquiries". On the right side of this bar, the text "janez(X Lab)" is displayed next to a downward arrow. Below this, a dropdown menu is open, showing three options: "Profile", "Change Company", and "Log Out".

Slika 5.3: Menu navadnega uporabnika, od koder se ta lahko tudi odjavi.

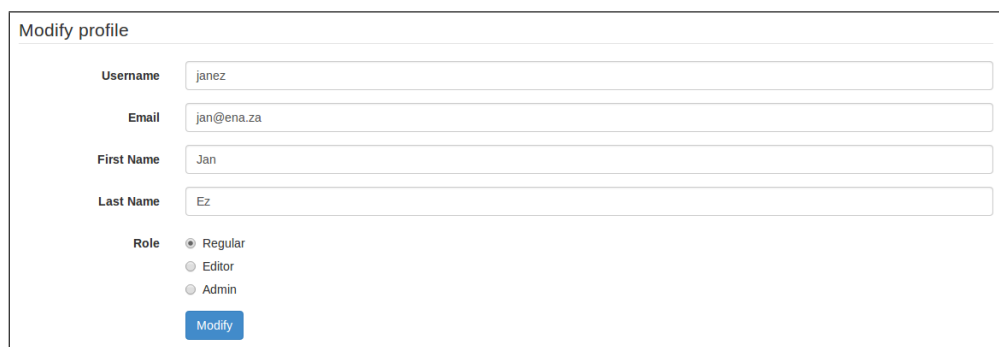
- Spreminjanje osebnih podatkov (Slika 5.4)



The screenshot shows a web form titled "Modify profile". It contains four text input fields: "Username" with the value "marko", "Email" with "ma@r.ko", "First Name" with "Mar", and "Last Name" with "Ko". The "First Name" field is currently selected with a blue border. Below the fields is a blue button labeled "Modify".

Slika 5.4: Obrazec za spremembo profila.

- Administracija uporabnikov (Slika 5.5)



The screenshot shows a web form titled "Modify profile" for an administrator. It includes the same four text input fields as Slika 5.4, but with different values: "Username" is "janez", "Email" is "jan@ena.za", "First Name" is "Jan", and "Last Name" is "Ez". Below these fields is a "Role" section with three radio buttons: "Regular" (which is selected), "Editor", and "Admin". A blue "Modify" button is at the bottom.

Slika 5.5: Obrazec za spremembo profila, do katerega lahko dostopa administrator.

- Pregled uporabnikov (Sliki 5.6 in 5.7)

The screenshot shows a web interface titled "All Profiles". At the top, there are navigation buttons: "← Previous" and "Next →", with a page indicator "1/1" in the center. Below this is a table with four columns: "Username", "Email", "First Name", and "Last Name". The table contains seven rows of user data. Below the table, there is another set of navigation buttons and a "Filter" button. Below the filter button is a form with four input fields: "Username", "Email", "First Name", and "Last Name".

Username	Email	First Name	Last Name
admin			
ancka	an@c.ka	An	Ckaa
francek	fra@nc.ek	Fra	Ncek
francka	fra@n.cka	Fran	Cka
janez	jan@ena.za	Jan	Ez
lojze	asd@asd.as	asd	asd
luka			

Slika 5.6: Pregled profilov in filtriranja uporabnikov.

The screenshot shows a web interface for a specific user profile, titled "janez". It displays the user's details: "Email" (jan@ena.za), "First Name" (Jan), and "Last Name" (Ez). Below the details are two blue buttons: "Change profile" and "Change password".

Email	jan@ena.za
First Name	Jan
Last Name	Ez

Slika 5.7: Podrobnosti profila določenega uporabnika.

- Filtriranje uporabnikov (Slika 5.6)
- Registracija podjetja (Slika 5.8)

The screenshot shows a web form titled "Register Company". It contains four input fields: "Company name", "Company Address", "Company Description" (a larger text area), and "Company Work Area". A blue button labeled "Register Company" is positioned at the bottom right of the form.

Slika 5.8: Registracija podjetja.

- Prijava in potrjevanje k podjetju (Sliki 5.9 in 5.10)

The screenshot displays a company profile for "Iskra". It lists the following details: Address (Kranj), Work Area (electricity elektrika električni števci), Leader (marko), Ratings (★★★★★★★☆☆ (8.67)), Jobs done (3), and Description (Company for electronical counters.). A blue button labeled "I work here" is located at the bottom right.

Slika 5.9: Pregled podjetja, ki ga vidi uporabnik, ki ne dela pod izbranim podjetjem.

Iskra

Address

Kranj

Work Area

electricity elektrika električni števeci

Leader

marko

Ratings

★★★★★★★★☆ (8.67)

Jobs done

3

Description

Company for electronical counters.

Modify settings

User	Can accept inquiry	Can apply inquiry	Can post inquiry	Can change company properties	Can grant company privileges	Is confirmed	Submit changes
marko	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<div>Submit</div>

Inquiries

Inquiry name

Zidanje

Tlakovanje

Pleskanje

Pleskanje

Offers

Inquiry name

Vrtnarjenje

Slika 5.10: Pregled podjetja, kot ga vidi prijavljen uporabnik s pristojnostmi za dodeljevanje pravic drugim uporabnikom.

- Spreminjanje lastnosti podjetja (Slika 5.11)

Modify Company

Company name: Iskra

Company Address: Kranj

Company Description: Company for electrical counters.

Company Work Area: electricity elektrika električni števci

Company Leader: marko

Modify Company

Slika 5.11: Obrazec za spreminjanje lastnosti podjetja.

- Seznam podjetij (Slika 5.12)

Companies

Create New Company

← Previous

1/6

Next →

Company name	Company address	Company work area
Akrapović	Ljubljana 2012	exhausts izpuhi
asd	asd	asd
Asd	Begunje 21, das	skies smuči
asd15	asd	asd
Boskarol	Ajdovščina 25	light planes lahka letala
Krka	Novo Mesto 23ab	farmacija pharmacy
Laško	Krško 5321	beer pivo
Peko	Tržič	dsa
Pixi	Šenčur	dsa
test	test	test

← Previous

1/6

Next →

Name

Company name

Work Area

Company Work Area

Address

Company Address


Filter

Slika 5.12: Pregled in filtriranje podjetij.

- Podrobnosti podjetja (Sliki 5.9 in 5.10)
- Iskanje podjetja (Slika 5.12)
- Objava povpraševanja (Slika 5.13)

Create Inquiry

Inquiry name

Inquiry final date 

Inquiry Work Area

Inquiry Details

Inquiry Area

Slika 5.13: Obrazec za objavo povpraševanja.

- Seznam povpraševanj (Slika 5.14)

All Inquiries

Create New Inquiry

← Previous

1/1

Next →

Inquiry name	Company name	Work area	Area	Time constraints
Building skyscraper	Petrol	building zidanje	Ljubljana	10.2.2015
Paving of city hall	Mestna občina Maribor	paving tlakovanje	Maribor	15.3.2015
Iron ore	Acroni	raw materials surovi materiali	Jesenice	10.12.2014

← Previous

1/1

Next →

Inquiry name

Inquiry name

Area

Area

Company name

Company name

Active date

Active date

Work Area

Work Area

Filter

Slika 5.14: Pregled in filtriranje povpraševanj.

- Iskanje povpraševanja (Slika 5.14)
- Podrobnosti povpraševanja (Slika 5.15)

Pleskanje

Time constraints

2014-10-23

Work Area

pleskanje

Details

90 kvadratnih metrov

Area

Celje

Company name

Iskra

Company Ratings

★★★★★★★★☆ (8.67)

Company done jobs

3

Modify inquiry

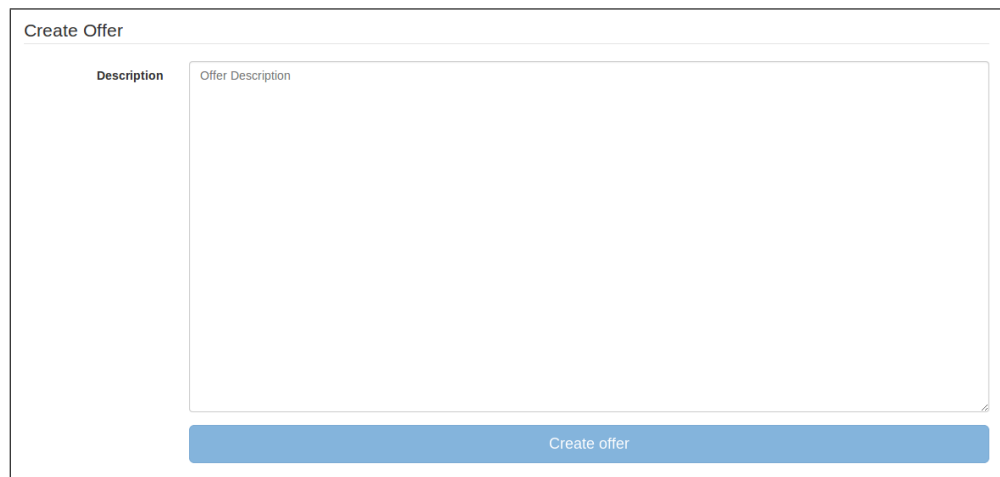
Offers

Company name

X Lab

Slika 5.15: Pregled podrobnosti povpraševanja.

- Prijava na povpraševanje (Slika 5.16)



The screenshot shows a web form titled "Create Offer". It features a large text area for the "Offer Description" with a label "Description" to its left. At the bottom right of the form is a blue button labeled "Create offer".

Slika 5.16: Obrazec za objavo ponudbe.

- Pregled ponudb (Slika 5.15)
- Podrobnosti ponudbe (Slika 5.17)

Offer from Iskra to X Lab (Inquiry: Vrtnarjenje)

Description

Sending the offer!

Modify offer

Offerer: Iskra

Company name

Iskra

Company Ratings

★★★★★★★ (8.67)

Company done jobs

3

Inquiry: Vrtnarjenje

Time constraints

2014-09-18

Inquiry work area

vrtnarstvo

Inquiry area

Ljubljana

Inquiry details

50 kvadratnih metrov za povrnariti

Inquirer: X Lab

Company name

X Lab

Company Ratings

☆☆☆☆☆☆☆☆ (0)

Company done jobs

0

Slika 5.17: Podrobnosti ponudbe.

- Spreminjanje ponudbe (Slika 5.18)

Modify Offer

Description

100 000 € pa naredim!

Modify Offer

Slika 5.18: Obrazec za spreminjanje ponudbe.

- Objava pogodbe(Sliki 5.19 in 5.20)

Contract between X Lab and Iskra (Inquiry: Pleskanje)

Description1000 € pa naredim!

Modify offer

Final contract submit

Offerer: X Lab

Company nameX Lab

Company Ratings☆☆☆☆☆☆☆☆ (0)

Company done jobs0

Inquiry: Pleskanje

Time constraints2014-10-23

Inquiry work areapleskanje

Inquiry areaCelje

Inquiry details90 kvadratnih metrov

Inquirer: Iskra

Company nameIskra

Company Ratings★★★★★★★★☆☆ (8,67)

Company done jobs3

Slika 5.19: Podrobnosti pogodbe.

Modify Contract

Description1000 € pa naredim!

Modify contract

Slika 5.20: Obrazec za spreminjanje pogodbe.

5.1.3 Pregled funkcionalnosti

Za boljši pregled nad izdelano funkcionalnostjo je narejena tabela z vsemi uporabniškimi zgodbami in stanjem zgodb po zaključenem diplomskem delu (Tabela 5.1). Stanja nam povejo, če je bila zgodba izdelana ali ne. Obstajajo tudi stanja delno, ki pomenijo delno izdelavo zgodbe. Takšno stanje vsebujejo Podrobnosti podjetja, Podrobnosti povpraševanja in Podrobnosti ponudbe. Pri vseh naštetih je izdelana večina funkcionalnosti, z izjemo dela, ki je povezan z uporabniško zgodbo Pogovor.

Uporabniška zgodba	Izdelano
Registracija uporabnika	Da
Prijava	Da
Odjava	Da
Pregled uporabnikov	Da
Filtriranje uporabnikov	Da
Spreminjanje osebnih podatkov	Da
Administracija uporabnikov	Da
Registracija podjetja	Da
Seznam podjetij	Da
Podrobnosti podjetja	Delno
Iskanje podjetja	Da
Prijava in potrjevanje k podjetju	Da
Spreminjanje lastnosti podjetja	Da
Objava povpraševanja	Da
Seznam povpraševanj	Da
Iskanje povpraševanja	Da
Podrobnosti povpraševanja	Delno
Prijava na povpraševanje	Da
Pregled ponudb	Da
Podrobnosti ponudbe	Delno
Spreminjanje ponudbe	Da
Objava pogodbe	Da
Ponujanje investicij	Ne
Seznam ponujenih investicij	Ne
Iskanje investicij	Ne
Pregled prijav	Ne
Podrobnosti prijav	Ne
Novice	Ne
O podjetju in kontakt	Ne
Pomoč	Ne
Pogovor	Ne

Tabela 5.1: Pregled uporabniških zgodb.

5.2 Varnost

O področju varnosti na naši aplikaciji je zaenkrat relativno težko govoriti, saj še nismo prišli do faze postavitve celotne aplikacije, ker ta še ni narejena. Načrtovano je, da bo aplikacija tekla preko zavarovane HTTP povezave prek interneta. Za tovrstno varnost bomo uporabili TLS plast.

Ker pa je nekoliko nesmiselno govoriti o tem kar je mišljeno za prihodnost, se bomo bolj osredotočil na del varnosti, za katerega je že poskrbljeno. Ker se morajo uporabniki prijaviti, je na mestu preverjanje avtentikacije. Ta je izvedena s pomočjo žetonov - JWT (na kratko so opisani v zadnjem odstavku opisa JSON-a - 3.3.2). Z njimi je za varnost poskrbljeno tako, da se prvi žeton s strežnika prenese k odjemalcu ob prijavi. Z izmenjavanjem žetonov ob nadaljni komunikaciji strežnik vedno ve, za katerega uporabnika gre, saj je vsakemu izmed njih dodelil svoj žeton. Ker ve za katerega uporabnika gre, posledično lahko tudi ugotovi kakšna pooblastila ima ta uporabnik, do katerih podatkov ima dostop in do katerih ne. Na strani odjemalca je pri tovrstni avtentikaciji potrebno hraniti žeton, ki ga je poslal strežnik. To je možno, kot pri seji (*session*), preko piškotkov, možno pa je tudi z uporabo lokalnega pomnilnika (*local storage*). Odločili smo se za drugo možnost.

Naslednji varnostni ukrep, ki se implementira sproti, je preverjanje podatkov, ki prispejo na strežniško stran preden se zapišejo v podatkovno bazo. Za podatke, ki prispejo od odjemalca moramo namreč vedeti, da so lahko kakršni koli, saj zahteve lahko spreminjajo navadni uporabniki. Tako se morajo vsi podatki validirati na strežniški strani. Ob pomoči Django REST Frameworka je validacija precej preprosta, saj moramo definirati le tip podatka na določenem mestu in po potrebi maksimalno dolžino.

Poglavje 6

Zaključek

Že po prvih pogovorih z Žigom Veselom, ki se je spomnil podjetniške ideje in k njenem razvoju prispeval domensko specifično znanje, smo vedeli, da je projekt preobsežen za celotno izdelavo v okviru diplomskega dela. Vseeno pa je bilo v začetku načrtovano, da bo izdelan večji del kot je bil. To je verjetno posledica tega, da določenih tehnologij, ki so uporabljene v projektu, pred tem še nikoli nismo uporabljali in jih je bilo tako potrebno najprej raziskati. Razlog za to je bilo verjetno tudi pomanjkanje časa, saj projekt do sedaj ni tekel tako gladko kot bi lahko.

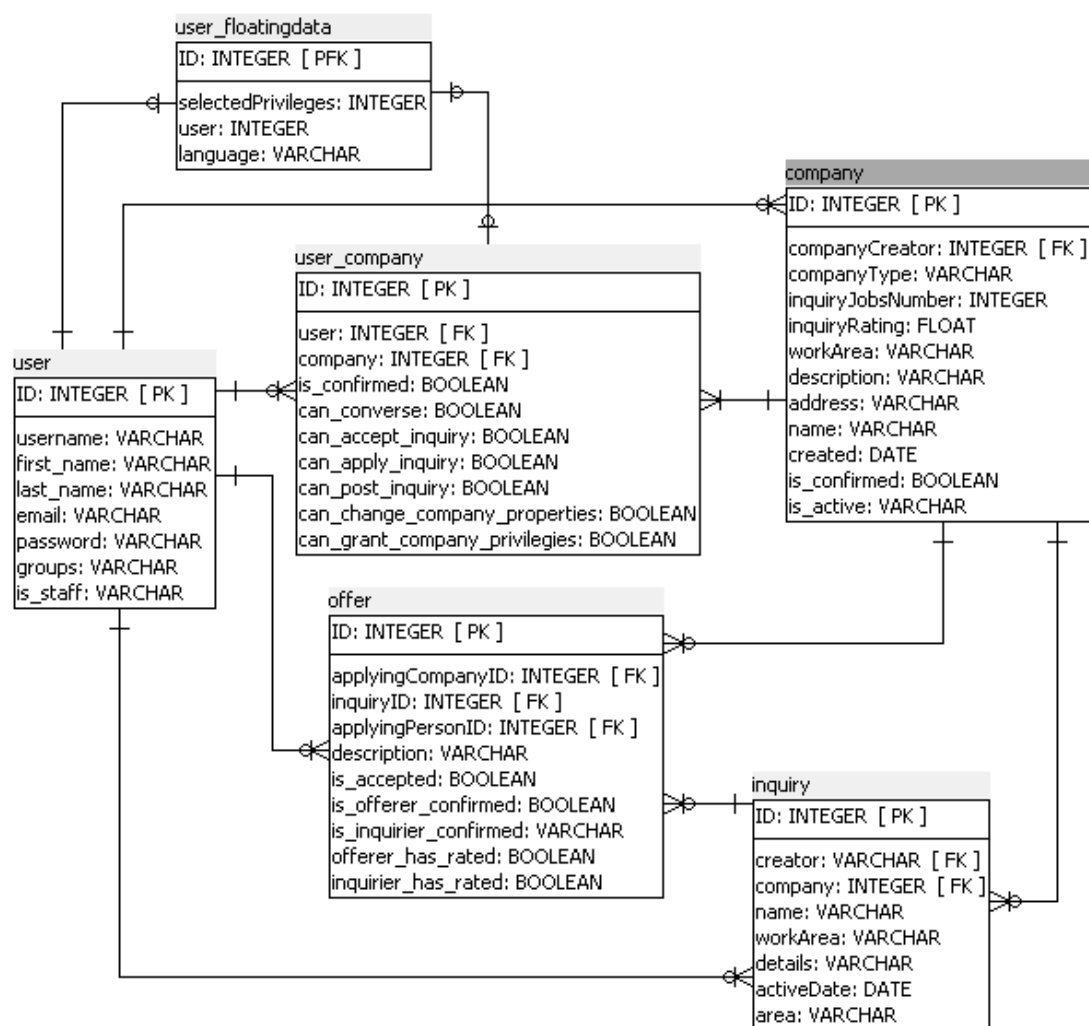
Vse stvari pa niso tako črne, kot se zdi. Izvedeno je namreč ogrodje spletne aplikacije, uporabljena tehnologija pa nam ni več nepoznana. Delo v prihodnje bo zato precej enostavnejše. Ravno prvi koraki v zasnovi so namreč najbolj zamudni, ko se je potrebno odločiti za tehnologijo in v njej napisati prve vrstice kode.

Projekt je šele nekje na polovici. Predvidevamo, da bo v prihodnosti tekel hitreje, saj izbrana tehnologija vsebuje precej manj pisanja kode, kot bi je bilo za isto funkcionalnost potrebno napisati v kakšnem drugem okolju. Določene stvari, ki so v izdelku izpuščene, so napisane v uporabniških zgodbah. Na kratko so to določene administrativne naloge, kot so urejanje novic, pomoči, o podjetju in pogovor, ter odstrani, ki se navezujejo na investicije. Te stvari bodo najverjetneje implementirane v bližnji prihodnosti, preden pa mislimo postaviti spletno stran, pa nameravamo sprogramirati še določene druge funkcionalnosti, kot so preverjanje uporabnika z uporabo certifikatov, možnost podpisa pogodbe na spletu, možnost

tiska pogodbe, po potrebi pa bomo dodali še kakšno drugo funkcionalnost.

Dodatek A

Podatkovni model podjetniške ideje



Literatura

- [1] Rodrigo Branas. Chapter 1. Getting Started with AngularJS. In *AngularJS Essentials*. Packt Publishing Ltd, 2014.
- [2] Gregor Capuder. *Določitev priporočene cene na spletni tržnici bolha.com*. PhD thesis, Univerza v Ljubljani, 2013.
- [3] Tom Christie. Django REST framework, 2014. Dostopano: 8.9.2014. Na voljo na spletu: <http://www.django-rest-framework.org/#django-rest-framework>.
- [4] Douglas Crockford. The application/json media type for javascript object notation (json), 2006. Dostopano: 8.9.2014. Na voljo na spletu: <http://www.rfc-editor.org/rfc/rfc4627.txt>.
- [5] John Deacon. Model-view-controller (mvc) architecture, 2009. Dostopano: 9.9.2014. Na voljo na spletu: <http://www.jdl.co.uk/briefings/MVC.pdf>.
- [6] Django Software Foundation. The Django template language, 2014. Dostopano: 8.9.2014. Na voljo na spletu: <https://docs.djangoproject.com/en/dev/topics/templates/>.
- [7] Roy Thomas Fielding. *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California, Irvine, 2000.
- [8] David Flanagan. Introduction to JavaScript. In *JavaScript: The definitive guide: Activate your web pages*. O'Reilly Media, Inc., 2011.
- [9] Jesse James Garrett and Others. Ajax: A new approach to web applications, 2005. Dostopano: 10.9.2014. Na voljo na spletu: ht-

- [tps://courses.cs.washington.edu/courses/cse490h/07sp/readings/ajax_adaptive_path.pdf](https://courses.cs.washington.edu/courses/cse490h/07sp/readings/ajax_adaptive_path.pdf).
- [10] Google. What Is Angular?, 2014. Dostopano: 8.9.2014. Na voljo na spletu: <https://docs.angularjs.org/guide/introduction>.
- [11] Adrian Holovaty and Jacob Kaplan-Moss. Chapter 1: Introduction to Django, 2009. Dostopano: 8.9.2014. Na voljo na spletu: <http://www.djangobook.com/en/2.0/index.html>.
- [12] Michael Jones, Paul Tarjan, Yaron Goland, Nat Sakimura, John Bradley, John Panzer, and Dirk Balfanz. JSON Web Token (JWT), 2012. Dostopano: 8.9.2014. Na voljo na spletu: <https://tools.ietf.org/html/draft-jones-json-web-token-09>.
- [13] Scott M Lewandowski. Frameworks for component-based client/server computing. *ACM Computing Surveys (CSUR)*, 30(1):3–27, 1998.
- [14] David Lucking-Reiley, Doug Bryan, Naghi Prasad, and Daniel Reeves. Pennies from ebay: The determinants of price in online auctions*. *The Journal of Industrial Economics*, 55(2):223–233, 2007.
- [15] Mark Lutz. Preface, A Python Q&A Session. In *Learning python*. O'Reilly Media, Inc., 2013.
- [16] Refsnes Data. HTML Introduction, 2014. Dostopano: 8.9.2014. Na voljo na spletu: http://www.w3schools.com/html/html_intro.asp.
- [17] Tutorial republic. Twitter Bootstrap Introduction, 2014. Dostopano: 8.9.2014. Na voljo na spletu: <http://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-introduction.php>.
- [18] Charles Severance. Java script: Designing a language in 10 days. *Computer*, 45(2):7–8, 2012.
- [19] W3C. HTML & CSS, 2013. Dostopano: 8.9.2014. Na voljo na spletu: <http://www.w3.org/standards/webdesign/htmlcss#whatcss>.

-
- [20] John Worsley and Joshua D Drake. What Is PostgreSQL? In *Practical PostgreSQL*. O'Reilly Media, Inc., 2002.